Pragas virtuais em Linux

VALGNEY CHERRI ISHIMI[1]
JOAQUIM QUINTEIRO UCHÔA[1]

[1]Curso ARL - DCC / UFLA - Cx Postal 3037 - CEP 37200-000 Lavras (MG)
valgney@yahoo.com.br,joukim@ginux.ufla.br

Resumo: *Este trabalho mostra que existem pragas virtuais que afetam o sistema operacional Linux e explica por que não há epidemia destas pragas. Mostra como se disseminam ou obtém privilégios de root e de que forma pode-se proteger o sistema. Apresenta uma experiência com pragas do Windows® emuladas no Linux e relaciona alguns produtos antivírus.*

Palavras-Chave: *Linux, vírus, worms, trojans, pragas.*

1 Introdução

O sistema operacional Linux está suscetível a ataques por pragas virtuais tanto quanto outros sistemas operacionais. Todavia, como são reportados poucas ocorrências de infecção desta natureza, prevalece a idéia de que o Linux seja imune a tais pragas. Em um importante tutorial Linux disponível na *web* encontra-se o seguinte trecho:

> O LINUX NÃO É VULNERÁVEL A VÍRUS! Devido a separação de privilégios entre processos e respeitadas as recomendações padrão de política de segurança e uso de contas privilegiadas (como a de root, como veremos adiante), programas como vírus tornam-se inúteis pois tem sua ação limitada pelas restrições de acesso do sistema de arquivos e execução.
>
> Frequentemente são criados exploits que tentam se aproveitar de falhas existentes em sistemas desatualizados e usa-las para danificar o sistema. Erroneamente este tipo de ataque é classificado como vírus por pessoas mal informadas e são resolvidas com sistemas bem mantidos. Em geral, usando uma boa distribuição que tenha um bom sistema de atualização resolve em 99.9% os problemas com exploits. (SILVA, 2005).

Textos como o transcrito sutilmente sugerem que o Linux seja imune a vírus. Mas é preciso que seja feita uma análise mais apurada dos fatos. Afinal, os vírus para Linux existem, embora em pequena quantidade. Assim, o objetivo deste trabalho é desmistificar essa idéia de imunidade, identificando até que ponto este ambiente é seguro contra pragas virtuais.

Um vírus de computador é um programa malicioso que adiciona seu código ao código de outro programa executável. Existem diversas outras pragas virtuais que comumente são confundidas com vírus, cada uma com características particulares, tais como:

***Trojan* (cavalo-de-tróia):** um programa que apresenta alguma funcionalidade útil, mas que esconde um código malicioso. Por exemplo, pode-se ter um joguinho que funciona corretamente, mas que de forma oculta executa algum código que rouba senhas ou altera as configurações do sistema;

***Worm*:** um programa cuja própria existência é oculta ao usuário. Normalmente, explora falhas de segurança no sistema operacional ou em outros programas, como servidores web ou de email. Não infecta outros programas, mas se dissemina distribuindo cópias suas para outros sistemas vulneráveis.

Mais detalhes sobre pragas virtuais podem ser verificadas em (UCHÔA, 2005). Neste documento, será utilizado o termo genérico "pragas virtuais" quando não for necessária uma distinção entre vírus e outros códigos maliciosos. É importante notar que seja qual for a forma e modo de agir da praga, representa um problema de segurança para o sistema.

Esta obra está organizada da seguinte forma: na Seção 2, é explicado por que não há epidemia de pragas no Linux. Na Seção 3, é mostrado como pragas podem realizar uma escalada de privilégios no sistema, sendo que alguns exemplos de pragas para Linux são relacionados na Seção 4. Com objetivo de aprofundamento, na Seção 5 são apresentadas análises de casos sobre ataques de *worms* ao servidor *web* Apache e sobre pragas do Windows® emuladas no Linux. Produtos antivírus e outras ferramentas de proteção são apresentadas na Seção 6. Por fim, na Seção 7 são apresentadas as principais conclusões a que foi possível chegar com este trabalho.

2 Ausência de epidemia

Uma das razões para não haver epidemia de vírus no Linux é que os vírus não sobrevivem ao ambiente hostil deste sistema (YEARGIN, 2005), onde a disseminação é dificultada pelo rígido controle de privilégios. Ou seja, se um vírus infectar a conta de um usuário, não conseguirá se propagar, não tendo grande repercussão. Este argumento só é válido quando a conta de superusuário (`root`) é utilizada com zelo, como em servidores e ambientes onde o usuário acesse sua estação de trabalho (*desktop*) sem privilégios de administrador.

Outro entrave à disseminação é a diversidade de plataformas de hardware e software em ambientes Linux (MOEN, 2005b). São muitos programas diferentes para navegar na web, clientes de email e de chat, editores de texto e multimídia, além da variedade de distribuições. Assim, se uma praga é escrita especificamente para explorar um falha em um determinado *software* com determinada versão e tal *software* não tiver grande representatividade, não haverá possibilidade de grande repercussão desta praga. Este fato deixará de ser um entrave à medida que alguns programas venham a se destacar em seus segmentos, como se observa com o OpenOffice.org como opção de ferramentas de escritório.

A postura dos desenvolvedores também dificulta a disseminação de pragas de aplicativos para Linux, quando optam pela segurança frente à comodidade. Por exemplo, os clientes de *email* no Linux não executam anexos diretamente. O anexo tem que ser salvo em disco e suas propriedades alteradas, para somente então ser executado pelo usuário.

A cultura do usuário na forma de utilizar o sistema operacional é um fator muito importante na disseminação ou não de pragas. Por exemplo, a boa prática recomenda que o uso da conta de administrador (`root`) seja restrita apenas às funções inerentes à

administração do sistema operacional. Essa boa prática tem sido largamente empregada no Linux. Com o aumento da utilização do Linux como estação de trabalho doméstica, contudo, acredita-se que este cenário mude, caso a conta `root` seja utilizada da mesma forma indiscriminada que a conta de administrador no Windows®.

Mesmo com todos os entraves apresentados, existe a possibilidade de disseminação de vírus no Linux. Basta imaginar um sistema onde exista um compartilhamento de arquivos e programas entre os usuários, comum em uma rede de computadores (PEREIRA; MAZZOLA, 2001). Se um usuário possuir acesso a dados compartilhados e inadvertidamente executar um vírus, este poderá infectar arquivos dos demais usuários que também tenham acesso ao compartilhamento.

A quantidade de pragas e infecções pode ser maior do que é divulgado. Por exemplo, a empresa Trend Micro publicou em seu *site* (`http://www.trendmicro.com`) que o *worm* Slapper infectou cerca de 1.219 servidores Apache. No entanto, analistas da empresa finlandesa F-Secure afirmam ter entrado na rede ponto-a-ponto criada pelo *worm* e contado a quantidade de servidores infectados, chegando ao número de 13,9 mil (RODRIGUES, 2002). Isso confirma as suspeitas de Radatti, que acredita que as estatísticas de ataque a ambientes Unix não são maiores porque os ataques realizados não são reportados (RADATTI, 2006).

Informações sobre novos incidentes e falhas podem ser consultadas nos *sites* `http://www.iss.net/` e `http://www.cert.org/`. A descoberta de uma nova praga pode ser comunicada à organização CERT (Computer Emergency Response Team) através do endereço eletrônico `cert@cert.org` ou à empresa ISS (Internet Security Systems) em `http://xforce.iss.net/xforce/contact`.

3 Pragas virtuais no ambiente Linux

O argumento mais utilizado para invocar a imunidade do Linux é que se uma praga virtual infectar uma máquina no perfil de um usuário comum, apenas os arquivos a que tal usuário tenha acesso estariam sujeitos às ações destrutivas daquela praga. Isso porque o Linux faz uso de instruções que requerem acesso privilegiado para realizar tarefas consideradas críticas ou que possam afetar todo o sistema.

Este argumento é falho, pois um vírus não necessita de poderes de root para infectar arquivos. Mesmo que o vírus esteja ativo com poderes normais de usuário, poderia obter privilégios através de diversas técnicas (RADATTI, 2006), como causar *buffer overflow* ou condição de corrida em algum serviço do sistema ou monitorar o teclado a espreita de alguma senha.

Buffer overflow é o resultado do armazenamento em um buffer de uma quantidade maior de dados do que sua capacidade. O princípio é estourar o *buffer* e sobrescrever parte da memória, alterando valores das variáveis locais, valores dos parâmetros e/ou endereço de retorno para que aponte para a área em que o código malicioso encontra-se armazenado. Apenas linguagens que não fazem checagem de limite são vulneráveis, como a linguagem C, na qual é baseado o *kernel* do Linux.

Ocorre condição de corrida quando múltiplos processos acessam e manipulam o mesmo dado concorrentemente e o resultado da execução possa ser diferente dependendo da

ordem em que tais acessos são feitos. Em algumas situações, uma falha deste tipo pode ser explorada para obter privilégios. Para exemplificar, foram reportadas no *site* `http://www.iss.net/` falhas no *kernel* 2.6 que permitiriam a uma praga virtual realizar uma escalada de privilégios:

- 12/01/2005: versões do *kernel* de 2.2 a 2.2.27-rc1, 2.4 a 2.4.29-rc1, e 2.6 a 2.6.10 permitiriam a um processo executando em nível de usuário obter privilégios elevados sobre o sistema, mediante uma condição de corrida em função de falha no gerenciador de paginação SMP (*Symmetric MultiProcessor*);
- 15/02/2005: versões do *kernel* anteriores a 2.6.11-rc4 permitiriam a um processo executando com privilégios *Direct Rendering Infrastructure* (DRA) obter privilégios elevados sobre o sistema, causado por uma condição de corrida no *driver* Radeon.

Outra forma de ganhar acesso privilegiado é roubando senhas. Desta forma, a utilização do comando `su` para obter privilégios de superusuário deve ser evitada. Um *keylogger*[1] pode estar alojado na conta de usuário e, se este eventualmente fizer uso do `su`, comprometerá a senha de `root`. Mais seguro, sob esse ponto de vista, seria efetuar *logoff* e se autenticar novamente como `root`.

Também é questionável a fácil aceitação que não seja um problema sério um usuário comum executar uma praga virtual em sua própria conta. Se o referido usuário, por exemplo, for o presidente de uma grande empresa e, de repente, seus documentos forem apagados por uma praga, isso causaria um grande transtorno. Mesmo que haja *backup* dos dados, deve-se levar em conta que pode haver uma defasagem entre o dado real perdido e o *backup*. Além disso, pode ser que a praga não apague dados, apenas facilite seu roubo ou adulteração.

Um vetor de disseminação de praga no Linux é a instalação de aplicações de terceiros, em boa parte das vezes como `root`. Seria simples criar algo que se parecesse com um programa legítimo (um *trojan*), como "DVD ripper e VCD encoder para Linux versão 2.34" (SEIFRIED, 2000). Muitos usuários baixam e instalam aplicativos inadvertidamente. Quando não funcionam, simplesmente desinstalam ou esquecem. De qualquer forma, a praga estaria instalada.

Além disso, os criadores de pragas virtuais utilizam-se da Engenharia Social para ludibriar o usuário. Tem-se como exemplo um falso *email* que circulou pela internet como sendo um aviso oficial da empresa Red Hat. O *email* continha instruções sobre como baixar e instalar uma "correção" para uma falha de *buffer overflow* nos comandos `ls` e `mkdir`. Na verdade, o *email* continha *links* para programas maliciosos, como apontado em (KOTADIA, 2004). Outra falsa idéia sobre o assunto é a crença de que seja difícil criar pragas para Linux. Pode ser verificado com facilidade que existem diversos documentos na Internet que ensinam como criar vírus para este sistema operacional, como (BARTOLICH, 2003).

[1] *Keylogger*: programa que monitora o teclado

4 Pragas virtuais para Linux

Até Junho de 2003, eram reportadas cerca de 100 pragas conhecidas para Linux. A partir de Novembro de 2003, este número pulou para 496, um aumento de 500%, conforme (KNIGHT, 2004). A Tabela 1 apresenta um breve histórico de algumas das pragas conhecidas para Linux, em ordem cronológica, tendo sido construída tomando-se por base informações disponibilizadas em diversas fontes. Algumas dessas pragas são consideradas "prova-de-conceito", ou seja, foram criadas com a finalidade de se demonstrar a corretude de um conceito. Dois conceitos que foram provados são a possibilidade de existência de vírus no sistema operacional Linux e a possibilidade de existência de vírus multiplataforma.

Pode-se encontrar pragas virtuais na forma de *scripts* ou códigos binários. Os *scripts* se beneficiam da portabilidade entre diferentes sistemas. A Figura 1 e a Figura 2 apresentam como pode ser uma infecção utilizando-se *scripts*. Assumindo-se que o vírus compreenda as nove primeiras linhas de código após a primeira linha, pode-se utilizar a seqüência de comandos apresentados na Figura 1 para embutir o código malicioso dentro de outro *script*. O código do *script* infectado ficaria de forma semelhante ao apresentado na Figura 2 (RADATTI, 2006).

Figura 1: Infecção por *script*

```
head -10 $0 > /tmp/trash
tail -9 /tmp/trash > /tmp/trash2
head -1 $target > /tmp/trash3
cat /tmp/trash3 /tmp/trash2 $target > /tmp/trash4
cat /tmp/trash4 > $target
/bin/rm -f /tmp/trash*
```

Figura 2: *Script* infectado

```
#!/bin/sh
[corpo do vírus]
#!/bin/sh
[código normal do script]
```

É importante comentar que em 30/07/2002, o servidor de FTP público da Fundação OpenBSD foi comprometido e um *trojan* foi adicionado ao fonte do pacote OpenSSH disponível àquela época. Isso foi descoberto e corrigido um dia depois. Outro incidente semelhante ocorreu em 28/09/2002 com o servidor de FTP público `ftp.sendmail.org`, onde foram adulterados os fontes do pacote do Sendmail 8.12.6, descoberto somente 8 dias depois (MOEN, 2005b). Outra informação curiosa é que, a fim de demonstrar que era possível ter um vírus multiplataforma, em 1994 havia sido criado um vírus que infectava documentos fonte do LaTeX, em sistemas que tinham o programa GNU Emacs (MCMILLAN, 1994).

Tabela 1: Breve histórico de pragas no Linux

Praga	**Descoberta**	**Descrição**
Bliss	Set/1996	Primeiro vírus conhecido para Linux. Considerado "prova-de-conceito". Seu código fonte fora disponibilizado nas listas `comp.security.unix`, `alt.comp.virus` e `comp.os.linux.misc` desde 29/09/96.
Staog	Out/1996	Alguns o consideram o primeiro vírus para Linux, pois o Bliss foi "descoberto" somente em fevereiro de 1997
ADM	Mai/1998	*Worm* composto de 8 arquivos. Infecta sistemas Linux através da exploração de uma falha do *daemon* `named` conhecida como *buffer overrun*. Este ataque consiste em enviar um pacote especialmente preparado à máquina-alvo do ataque, que é executado e causa uma excessão que lhe dá poderes de `root`. O arquivo `/etc/hosts.deny` é excluído e é aberta uma conexão com a máquina de origem por onde é baixado o restante do código do *worm*. Quando a infecção se completa, o *worm* envia um *email* para `admsmb@hotmail.com` e inicia um novo ataque a outro sistema vulnerável.
Ramen	Jan/2001	Este *worm* utiliza vulnerabilidades em 3 *daemons* do RedHat 6.2 e 7 que lhe garantem poderes de `root`: `statd`, `lpd` e `wu-ftp`. As vulnerabilidades foram reportadas 3 meses antes do *worm* ser descoberto. É composto de 26 arquivos, dentre *scripts* e executáveis.
Lion	Mar/2001	Este *worm* explora a vulnerabilidade *Transaction Signatures Buffer Overflow*, que permite a execução de código arbitrário no sistema atacado. Envia certas informações para um dado endereço de *email*, como configuração da rede e arquivos `passwd` e `shadow`.
Winux	Mar/2001	É um vírus multiplataforma, pois infecta Windows® 95, 98, ME, NT e 2000, além do Linux. É considerado "prova-de-conceito".
Adore	Abr/2001	Similar ao Ramen e Lion, este *worm* utiliza vulnerabilidades conhecidas nos serviços BIND, `wu-ftpd`, `rpc-statd` e `lpd` para infectar o sistema. Uma vez infectado, faz *download* do *worm* e o executa. Depois instala um *backdoor* (porta de acesso não autorizado ao computador, quando este está conectado à Internet).
Slapper	Set/2002	Este worm explora uma vulnerabilidade do servidor *web* Apache com OpenSSL (`mod_ssl`) versão 0.96d e anteriores. Ele enviava uma requisição HTTP GET inválida na porta TCP 80. Quando detectado um sistema vulnerável, conectava-se ao serviço SSL na porta TCP 443, enviava uma cópia de seu código-fonte, compilava e executava.
Mighty	Out/2002	Similar ao Slapper, este *worm* explora uma falha de *buffer overflow* no Apache executando OpenSSL versões 0.9.6d, 0.9.7-beta2 e anteriores, obtendo privilégios de `root`.
Troj_Remoterx.A	Out/2003	*Trojan* que utiliza *buffer overflow* para acessar e controlar sistemas Linux , Windows® NT, 2000 e XP.
Binom	Dez/2004	Nos arquivos infectados por este vírus pode-se observar o texto: "`[ Cyneox/DCA`".

5 Casos Clássicos

5.1 *Worms* no servidor *web* Apache

O Apache é um servidor *web* desenvolvido de forma colaborativa e disponibilizado com código aberto, inicialmente baseado no HTTPD escrito pelo NCSA. Tem sido o mais popular servidor *web* desde abril de 1996, chegando a abranger, em novembro de 2005, quase 70% dos servidores *web* no mundo, segundo (NETCRAFT, 2005). Existem versões do Apache para diferentes Sistemas Operacionais, como Linux e Windows®. Devido a sua grande popularidade, tem sido alvo de ataque de *worms* como Scalper e Slapper.

O Scalper foi inicialmente detectado em Junho de 2002. Esse worm propagava-se em ambientes FreeBSD pela existência de uma vulnerabilidade do servidor *web* Apache, conhecida como *encoding vulnerability*. Essa vulnerabilidade foi eliminada a partir das versões 1.3.26 e 2.0.39 do Apache (TOCHEVA; RAUTIAINEN, 2002).

Quando o Scalper ganha acesso ao servidor, copia um arquivo com extensão `.uaa` na pasta `/tmp`, decodifica-o, gerando um novo arquivo com extensão `.a`, que é executado. Ele permite a um atacante controle remoto sobre o servidor comprometido através de um *backdoor* na porta 2001/UDP, com os mesmos privilégios do servidor Apache, e inicia um novo ataque a outras redes classe A. A remoção deste *worm* é simples, bastando excluir o executável na pasta `/tmp` e terminar sua execução. O seu processo pode ser visto na lista de processos como "`.a`" e finalizado com o comando `kill`. O Scalper não modifica a configuração do sistema.

O Slapper apareceu três meses depois do Scalper, muito semelhante ao primeiro. Seu alvo eram máquinas executando Apache e OpenSSL (`mod_ssl`) anterior a 0.96d, inclusive. Todavia, ao contrário do Scalper, o Slapper atingiu milhares de servidores *web* pelo mundo, atacando sistemas Unix, Solaris, Linux e BSD compatíveis. Embora existam versões do Apache para Windows®, este não fora afetado. Como comentado em (RODRIGUES, 2002):

> As falhas exploradas pelo Slapper encontram-se na biblioteca de arquivos do OpenSSL, uma implementação de código aberto do protocolo SSL (Secure Sockets Layer), usado para transações seguras pela Internet, principalmente por sites de comércio eletrônico, bancos e aplicações que requerem privacidade

O Slapper procurava por sistemas vulneráveis enviando uma requisição HTTP GET inválida na porta TCP 80. Quando detectado um sistema Apache vulnerável, o *worm* se conectava ao serviço SSL na porta TCP 443, enviava uma cópia de seu código fonte (`.bugtraq.c`) para a pasta `/tmp`, decodificava o código-fonte e o compilava, gerando o executável `.bugtraq`, que era então executado. O *worm* ativo no servidor comprometido iniciava uma varredura para infectar outros servidores (CERT, 2002).

Cada servidor infectado pelo Slapper passava a fazer parte de uma espécie de rede ponto-a-ponto criada pelo *worm*, que poderia ser manobrada pelo criador do vírus para realizar ataques do tipo DDoS (*Distributed Denial of Service*). Ataques deste tipo ocorreram contra empresas como Amazon, Yahoo, CNN, eBay e Microsoft. O Slapper infectou

cerca de 13,9 mil servidores, segundo a empresa finlandesa F-Secure, que chegou a estes números penetrando na rede ponto-a-ponto criada pelo worm e contando os servidores comprometidos.

Considera-se ainda duas variantes principais do Slapper: Cinik e Unlock. Exploravam a mesma vulnerabilidade, mas manuseavam arquivos com nomes distintos, além de corrigirem falhas de programação de seu antecessor. A variante Unlock ainda adicionou a capacidade de instalar um *backdoor* na máquina infectada, que ficava monitorando a porta 1052/TCP.

O Slapper e suas variantes geram um grande volume de tráfego nas portas 2002/UDP, 1978/UDP e 4156/UDP, degradando o desempenho de redes e comprometendo a resposta dos servidores. O código-fonte do Slapper pode ser encontrado para estudo no endereço `http://packetstormsecurity.org/0209-exploits/bugtraqworm.tgz`.

5.2 Pragas do Windows® no Linux

Em função da diferença entre arquiteturas, as pragas virtuais que atingem o Windows® geralmente não causam danos ao Linux. Abre-se uma exceção a pragas multiplataforma como Winux, que atinge ambos os sistemas. Obviamente, O processo de boot do Linux pode ser interrompido por um vírus do Windows® que altere áreas como o Master Boot Record (MBR), em máquinas com *dual boot*.

Quando uma praga virtual do Windows® é executada no Linux a partir de emulação com o Wine, é possível que tal praga consiga causar algum dano, conforme mostrado na Tabela 2, adaptada de (MOEN, 2005a).

Tabela 2: Execução de pragas do Windows® no Linux

	Klez	**MyDoom**	**Sobig**	**SCO**	**SomeFool**
Executa	Sim	Não	Sim	Sim	Sim
Instala Payload	Não	Não	Não	Sim	Sim
Propaga-se	Não	Não	Não	Não	Não
Afeta Linux	Não	Não	Não	Não	Sim

Segue-se uma uma análise detalhada da execução das pragas virtuais do Windows®, através do Wine, apresentadas na Tabela 2, de acordo com testes apresentados em (MOEN, 2005a):

- O Klez executa, mas o Wine emite mensagens de erro acerca do arquivo `ntdll`. Este *worm* procura por endereços de *email* e envia *emails* com sua cópia em anexo. Foi adicionado um endereço de *email* a um arquivo .txt em `~/wine/fake_windows/Windows/Desktop/` e reexecutado o *worm*; mesmo assim nenhum *email* foi recebido referente ao *worm*.
- MyDoom se dissemina a partir de um arquivo compactado `.zip` . Como não foi possível descompactar, este *worm* foi considerado incompatível com Linux/Wine.
- Sobig supostamente cria o arquivo *winstt32.dat* ao ser executado e envia *emails* com cópias suas. No Wine, executou mas não criou o tal arquivo nem enviou *emails*.

- SCO é uma variante do MyDoom, mas ao contrário de seu predecessor, foi possível descompactar seu arquivo `.zip` . Ao ser executado, conseguiu gravar seu *payload* em `~/wine/fake_windows/Windows/System/shimgapi.dll` e alterou a data do sistema para 03/02/2004.
- SomeFool conseguiu instalar seu arquivo `winlogon.exe` no Wine e entrou em *loop*, impactando negativamente na performance da máquina.

Com o aperfeiçoamento do Wine e o surgimento de pragas multiplataforma como o Winux, pode ser que em um futuro próximo estes resultados sejam mais favoráveis às pragas virtuais. Há de se considerar também uma boa probabilidade de vírus de macro do MS Office® executar em Linux via Wine. Outra possibilidade é a execução de vírus de macro do OpenOffice.org.

6 Ferramentas para combate às pragas virtuais

6.1 Produtos antivírus

Como atualmente não existem muitas pragas para Linux, os programas antivírus que executam neste sistema possuem sua base de assinaturas quase que totalmente voltadas para pragas do Windows®. Uma assinatura é uma seqüência de caracteres utilizada para identificar uma praga.

O leitor poderia questionar sobre o motivo procurar pragas do Windows® no Linux. A resposta é simples: o Linux é muito utilizado em servidores de rede (servidor de email, de dados etc). Assim, supondo-se que um servidor de emails no Linux sirva estações Windows®, seria desejável que as mensagens com anexo fossem verificadas antes de serem entregues aos clientes.

Também é interessante ter antivírus na estação de trabalho Linux, pois mesmo que o antivírus relacione apenas uma centena de pragas do Linux, ainda assim pode detectar uma praga desconhecida utilizando-se de heurísticas e análise de padrões, como apontado em (VELASCO, 2004). Alguns produtos antivírus para Linux são:

- F-Prot® Antivirus for Linux x86 Workstations[2]: protege estações Linux da ação de vírus, *worms* e *trojans*. Pragas desconhecidas podem ser detectadas através de heurística. Executa a partir de linha de comando. É grátis para uso pessoal, sendo que a versão paga inclui um componente de monitoramento (*Real Time Protector*), um agendador de verificação automática e um gerenciador de atualizações de assinatura.
- McAffe® LinuxShield[3]: programa comercial que protege o sistema Linux contra vírus, *worms*, *trojans* e outras pragas. Possui mecanismo de atualização automática e implementa heurística.
- BitDefender® Linux Edition[4]: programa gratuito executado por meio de linha de comando. Elimina *worms* como Morris e Scalper, além de pragas do Windows.

[2]F-Prot: `http://www.f-prot.com/`.
[3]McAffe: `http://www.mcafeesecurity.com/`.
[4]BitDefender: `http://www.bitdefender.com/`.

- Clam Antivirus (ClamAV)[5]: programa distribuído sob a licença GPL (Licença Pública Geral). Seu propósito principal é o uso integrado com servidor de *email* (verificação de anexos), mas pode ser utilizado para verificar o sistema de arquivos de uma estação. Possui uma ferramenta para atualização automática. Executa por linha de comando, mas existem ferramentas de terceiros que podem lhe conferir uma interface gráfica, como o ClamShell.

6.2 Prevenção contra *buffer overflow* e condição de corrida

Como comentado na Seção 2, um vírus poderia se aproveitar de falhas em programas, e buscar ampliar seus privilégios através da exploração de *buffer overflow* ou condição de corrida, por exemplo. Dessa maneira, torna-se recomendável o uso de ferramentas que previnam esse tipo de falha, como RootCheck, Libsafe e PaX.

O RootCheck[6] é uma ferramenta que atua quando um usuário comum muda sua *uid* para 0 (`root`), verificando se este usuário tem permissão para tal. Em caso negativo, o RootCheck derruba o processo iniciado pelo usuário. Desta forma, se uma praga virtual utilizasse uma vulnerabilidade de *buffer overflow* para conseguir privilégios de root, seu processo seria derrubado pelo RootCheck, como apontado em (LEON, 2003), o mesmo se aplicando à condição de corrida.

A Libsafe[7] é uma biblioteca de carregamento dinâmico que protege os processos contra explorações de *buffer overflow* pela interceptação de chamadas a determinadas funções conhecidas como vulneráveis, como `strcpy(char *dest, const char *src)` e `gets(char*s)`, consoante (TAMBORIM, 2004). É executada uma versão substituta da função interceptada, que implementa a mesma funcionalidade mas que não está sujeita a esse tipo de falha.

O PaX[8] é um *patch* para o *kernel* do Linux que engloba vários mecanismos de defesa contra exploração de falhas que permitiriam a um atacante obter acessos arbitrários de leitura/escrita no espaço de endereçamento do processo atacado. O PaX previne ataques por *buffer overflow*.

7 Conclusão

Existem muitas vozes que apontam o crescimento da quantidade de pragas para Linux à medida que este sistema aumente sua participação no mercado de estações de trabalho. Tais perspectivas podem ou não se concretizar. O importante é notar se tais pragas terão algum impacto no mundo Linux. Esse impacto dependerá dos seguintes fatores :

Cultura dos usuários: a cultura atual dos usuários Linux, talvez pelo grande emprego em universidades e empresas, beneficia o sistema. Com o aumento do uso do Linux em estações de trabalho por usuários domésticos, muitos dos quais migrando do Windows® para Linux, esse panorama pode mudar, muito embora o próprio sistema e programas afins estimulem o usuário a boas práticas. Por exemplo, ao executar o

[5] ClamAV: `http://www.clamav.net`.
[6] RootCheck: `http://www.w00w00.org/files/sectools/rootcheck/`.
[7] Libsafe: `http://www.research.avayalabs.com/gcm/usa/en-us/initiatives/all/nsr.htm`.
[8] PaX: `http://pax.grsecurity.net`.

cliente de IRC X-Chat como `root`, o aplicativo emite uma mensagem informando que essa é uma ação perigosa e sugere que utilize o serviço com uma conta de usuário.

Postura dos desenvolvedores: os desenvolvedores Linux em geral dão muito valor à segurança. Atualmente, por exemplo, existe um repúdio à execução automática de anexos em *emails*. Se futuramente os desenvolvedores abrirem mão da segurança em prol da comodidade, poderão criar um ambiente propício à disseminação de pragas, como ocorre com o Windows®.

O grande problema quanto a pragas no Linux não são os vírus, mas sim os *worms* e *trojans*. O Linux não é imune a pragas virtuais. Seria mais apropriado dizer que atualmente está menos propenso a ataques que outros sistemas operacionais. A segurança precisa ser construída. Deve-se manter o sistema operacional e aplicativos atualizados, aperfeiçoar suas configurações, utilizar ferramentas de auditoria e criar uma cultura de boas práticas entre os usuários.

Em ambientes controlados e atualizados, como servidores, onde se utiliza a conta `root` com cautela e não se executa programas suspeitos, não se justifica o emprego do antivírus, a não ser que o servidor trabalhe com clientes Windows®. Uma alteração não autorizada em tal sistema seria mais facilmente detectada e corrigida. Em estações de trabalho, onde não haja tal rigor e a conta root seja utilizada de forma abusiva, como em estações domésticas, seria prudente o emprego de antivírus.

Desta forma, fica claro ao administrador de sistemas Linux que ao se pensar em segurança não se deve deixar de lado a preocupação com pragas virtuais e que é muito importante que seus usuários recebam orientações sobre boas práticas, como apontado em (KOJM, 2006). Nesse sentido, com o objetivo de ampliar o leque dessas prática, este trabalho sugere como trabalhos futuros a construção de ambiente com HoneyPot[9] para pragas virtuais, com testes das várias ferramentas de antivírus. Com isso, seria possível identificar pragas de Linux ativas e a eficiência dos produtos antivírus.

Referências

BARTOLICH, A. *The ELF Virus Writing HOWTO*. 15 fev. 2003. WWW. Disponível em: <http://www.linuxsecurity.com/resource_files/documentation/virus-writing-HOWTO/_html/index.html>. Acesso em: 19/12/2005.

CERT. *CERT Advisory CA-2002-27 Apache/mod_ssl Worm*. 11 out. 2002. WWW. Disponível em: <http://www.cert.org/advisories/CA-2002-27.html>. Acesso em: 19/12/2005.

KNIGHT, W. *Pundits predict malware may target Linux*. 11 nov. 2004. WWW. Disponível em: <http://www.newsforge.com/article.pl?sid=04/11/10/2014240>. Acesso em: 19/12/2005.

KOJM, T. É seguro mesmo?: O caminho da infecção no Linux. *Linux Magazine*, n. 17, p. 23–6, fev. 2006. Disponível em: <http://www.linuxmagazine.com.br/issue/17>.

KOTADIA, M. *Linux under threat from 'security update'*. 25 out. 2004. WWW. Disponível em: <http://news.zdnet.co.uk/internet/security/0,39020375,39171264,00.htm>. Acesso em: 19/12/2005.

[9] http://www.honeypots.net/

LEON, N. *Passando por rootcheck*. 30 abr. 2003. WWW. Disponível em: <http://www-.securityexperts.com.br/modules.php?name=Content&pa=showpage&pid=10>. Acesso em: 19/12/2005.

MCMILLAN, K. A. *A Platform Indenpendent Computer Virus*. Dissertação (Master's Thesis) — University of Wisconsin, Milwaukee, 1994. Disponível em: <http://www.users.qwest.net-/~eballen1/virefs.html>.

MOEN, M. *Running Windows viruses with Wine*. 26 jan. 2005. WWW. (NewsForge). Disponível em: <http://os.newsforge.com/article.pl?sid=05/01/25/1430222>. Acesso em: 19/12/2005.

MOEN, R. *Rick's Rants*. 13 dez. 2005. WWW. Disponível em: <http://linuxmafia.com/~rick/faq-/index.php?page=virus>. Acesso em: 20/02/2006.

NETCRAFT. *Web Server Survey Archives*. 7 nov. 2005. WWW. Disponível em: <http://news-.netcraft.com/archives/web_server_survey.html>. Acesso em: 19/12/2005.

PEREIRA, K.; MAZZOLA, V. B. Vírus e malware em ambientes Unix e Linux. In: CTA/ITA. *Anais do III Simpósio Segurança Em Informática SSI'2001*. São José dos Campos, 2001. p. 251–8. Disponível em: <http://linorg.cirp.usp.br/SSI2001/artigos.html>.

RADATTI, P. V. *The Plausibility of UNIX Virus Attacks*. abr. 2006. WWW. (Cybersoft White Papers). Disponível em: <http://www.cybersoft.com/whitepapers/papers/plausibility.shtml>. Acesso em: 19/12/2005.

RODRIGUES, G. *Novo vírus para Linux infecta milhares de servidores*. 16 set. 2002. WWW. Disponível em: <http://www.infoguerra.com.br/infonews/talk/1032183241,36036,.shtml>. Acesso em: 19/12/2005.

SEIFRIED, K. *Linux Viruses: Overview*. 5 dez. 2000. WWW. Disponível em: <http://www-.developer.com/open/article.php/625211>. Acesso em: 19/12/2005.

SILVA, G. M. da. *Guia Foca GNU/Linux - Versão Iniciante - 3.99 - Capítulo 1 - Introdução*. 2005. WWW. Disponível em: <http://focalinux.cipsga.org.br/guia/iniciante/index.htm>. Acesso em: 19/12/2005.

TAMBORIM, A. L. *Libsafe: protegendo Linux contra smashing overflows*. 6 dez. 2004. WWW. Disponível em: <http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=1577>. Acesso em: 19/12/2005.

TOCHEVA, K.; RAUTIAINEN, S. *F-Secure Virus Descriptions : Scalper*. 29 jun. 2002. WWW. (F. Secure Corporate). Disponível em: <http://www.f-secure.com/v-descs/scalper.shtml>. Acesso em: 19/12/2005.

UCHÔA, J. Q. *Segurança Computacional*. 2. ed. Lavras: UFLA/FAEPE, 2005. (Curso de Pós Graduação "Lato Sensu" (Especialização) a Distância em Administração em Redes Linux).

VELASCO, M. Técnicas para detecção de vírus. In: CARMONA, T. (Ed.). *Guia Hacker*. São Paulo: Digerati Books, 2004. p. 60–5.

YEARGIN, R. *The short life and hard times of a Linux virus*. 30 jul. 2005. WWW. Disponível em: <http://librenix.com/?inode=21>. Acesso em: 19/12/2005.