SISTEMA PARA DETECÇÃO DE *EXPLOITS* NO LINUX

Fernando Leite Barreto[1]; Fabrício Sérgio de Paula[2];

[1] Estudante do Curso de Ciência da Computação da UEMS, Unidade Universitária de Dourados; e-mail: nando_leite.barreto@hotmail.com

[2] Professor do Curso de Ciência da Computação da UEMS, Unidade Universitária de Dourados; email: fabricio.paula@gmail.com

Área temática: Redes de Computadores e Segurança Computacional

Resumo.

Exploit é um programa, comando ou sequência de dados que usufrui da vulnerabilidade de um sistema, objetivando invadi-lo. Normalmente, tem o intuito de prever as falhas do sistema para futuras correções, mas em grande parte dos casos, seu escopo é muito menos nobre.

Muitas vezes, um e*xploit* entra no computador interpretado como dados inofensivos. Estes dados, no entanto, provocam a instabilidade do sistema para diminuir temporariamente sua segurança. Assim, ele passa a executar ordens em seu sistema para roubar informações, invadir acessos bloqueados ou enviar um vírus.

Devido isso, a proposta deste projeto é continuar a análise *exploits* para aplicações que executem sob o sistema operacional Linux. Pretende-se utilizar os estudos realizados sobre ações desenvolvidas por *exploits* durante um ataque de forma a elaborar um modelo e protótipo de sistema de detecção de intrusão (IDS).

Palavras chave:

Segurança computacional. Sistemas Operacionais. Sistema de detecção de intrusão.

1 – **Introdução.**

A excessiva utilização da Internet tornou-se um hábito na vida das pessoas. Porém, a mesma evolução tecnológica que nos proporciona uma série de benefícios, traz consigo transformações que atingem de maneira significativa seres humanos que esperavam “colher apenas bons frutos” à sociedade, desconhecendo o lado sombrio da sociopatia, o desejo de inserir o mal na vida das pessoas.

Diversas tecnologias são empregadas focadas a segurança computacional, incluindo *firewalls*, mecanismos criptográficos e sistemas de detecção de intrusão (*Nakamura* & *de Geus*, 2003). Sendo assim, um IDS (Sistema de Detecção de Intrusão) é uma ferramenta eficiente, devido ter como principal objetivo a identificação de

ocorrências de ataques em computadores ou redes, visando não comprometer o sistema atacado.

Este trabalho tem a proposta de uma continuidade prévia sobre a análise *exploits*, que são ataques projetados para explorar fraquezas em aplicações de modo a comprometer o sistema atacado (Northcutt et al., 2001). A análise de *exploits* para aplicações Linux[1] possibilita entender qual o comportamento inicial de processos atacados nesse tipo de sistema. Destarte, busca-se a elaboração de um modelo e um protótipo de IDS para identificar a ação de *exploits* em um sistema Linux.

2 – **Materiais e Métodos**

As fontes bibliográficas incluem artigos disponíveis na Internet e de obras disponíveis nas bibliotecas da UEMS e UFGD.

As atividades práticas foram realizadas através do uso de máquinas virtuais. Foi utilizado o sistema operacional Linux, *kernel* 2.6, e ferramentas/bibliotecas de código aberto e gratuito, de forma que nos termos dos requisitos da pesquisa, os mesmos foram devidamente adaptados.

3 – **Resultados/Discussão**

Nesta seção, são mostradas algumas definições e conceitos importantes referentes ao assunto tratado.

3.1 – **Segurança**

De fato, a Internet se tornou um meio de acesso indispensável na vida das pessoas, onde os mesmos se conectam em um mundo virtual repleto de informação e conhecimento, porém, práticas ilegais vêm se tornando comum em nosso cotidiano.

Segundo Tanenbaum (2003, p.768), a segurança é um assunto abrangente que se preocupa em garantir que usuários com más-intenções não invadam a privacidade de outros, a fim de evitar maiores transtornos e proteção.

3.2 – **Vulnerabilidade**

De acordo com Anderson (2005), vulnerabilidade é um erro de programação encontrado em projetos ou mesmo na criação de um *software*, o qual, uma vez explorado por um atacante, viola a segurança do usuário.

Na figura 3.1, Junior (2010) mostra que o numero de ataques a sites web vem crescendo de forma desequilibrada, com a tendência de ascender cada vez mais.

[1] Detalhes do sistema operacional Linux podem ser encontrados em http://www.kernel.org.

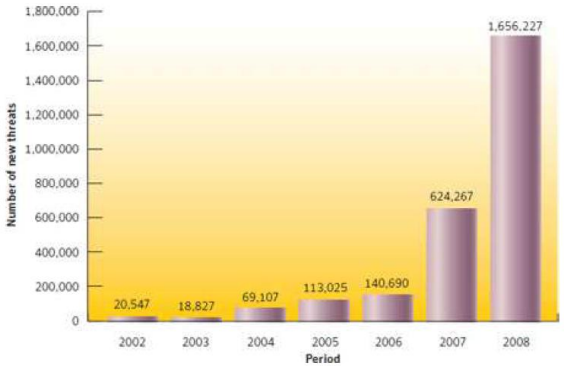

Figura 3.1: Volume de ataques a sites da *web*

3.3 – **Máquina Virtual**

Segundo Laureano (2006), uma máquina virtual (*Virtual Machine – VM*) é uma duplicata eficiente e isolada de uma máquina real, realizando uma cópia isolada de um sistema físico, a qual é totalmente protegida.

A *software* utilizado para criar a máquina virtual foi o *Oracle VM Virtualbox*[2] versão 4.0.8. O sistema operacional escolhido para ser instalado na máquina virtual foi o *BackTrack*[3] 5 na versão 32 *bits* e 64 *bits*.

Segundo Renato (2012), o *Backtrack* é um sistema operacional Linux focado para testes de segurança e penetração (*pen tests*), bastante utilizados por *hackers* com fins menos nobres e analistas de segurança, a fim de avaliar a segurança dos sistemas de computação e/ou rede, simulando um ataque. Possui diversas ferramentas destinadas a testes de penetração em sistemas, verificação de vulnerabilidades, na utilização de avaliação, dentre outras.

3.4 – ***Exploit***

Conforme Almeida (2008), o termo *exploit* se refere a pequenos códigos de programas desenvolvidos especialmente para explorar falhas introduzidas em aplicativos por erros involuntários de programação.

Uchôa (2005) alerta sobre a importância de se manter sempre muito bem informado sobre falhas em serviços e disponibilização de novas correções em sites especializados para evitar a ação dos *exploits*. A atualização do servidor deve ser feito constantemente assim que forem disponibilizadas as devidas correções.

3.5 – **IDS (*Intrusion Detection System*)**

Segundo Igreja (2007), o IDS (Sistema de Detecção de Intrusos) é um sistema que detecta vários tipos de tráfego da rede (NIDS) e também de computadores

[2] Detalhes do *software Oracle VM Virtualbox* podem ser encontrados em http://www.virtualbox.org/

[3] Detalhes do sistema operacional *Backtrack* podem ser encontrados em http://www.backtrack-*Linux*.org/

maliciosos (HIDS) que não são descobertos pelo *firewall* da rede.

3.6 – **Chamada de sistema**

Segundo Cardozo (2008), é um método utilizado pelo processo para realizar um requerimento a um determinado serviço do sistema operacional, ou mais especificamente, ao *kernel*.

3.7 – ***Strace***

Segundo Cardozo (2008), a função do *strace* é rastrear as chamadas de sistema produzidas por um determinado processo, visando colher informações sobre um determinado processo. Essas informações servem de grande valor não só para correções de problemas do sistema, ou até mesmo, na identificação de um ataque.

4 - Monitoramentos de intrusões e chamadas ao sistema operacional

Nesta seção, é mostrado um modelo de como os processos devem ser analisados, ilustrado na figura 4.1.

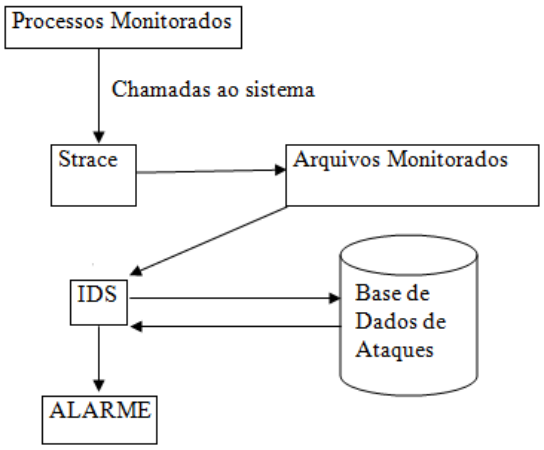

Figura 4.1: Modelo da análise de um processo feito pelo IDS baseado em máquina

A fim de verificar intrusões os processos foram analisados através das chamadas ao sistema feitas pelos mesmos, as quais, segundo *Silberschatz* (2004), foram classificadas em cinco categorias principais: controle de processo, gerenciamento de arquivo, gerenciamento de dispositivo, manutenção de informação e comunicações.

Para identificar cada processo, foi utilizado o PID (*Process Identifier*). A coleta das informações sobre as chamadas feitas pelo processo será feita pela ferramenta *strace*, a qual possui diversas formas de execução, onde poderá ser visualizada pelo comando: *man strace*

O comando utilizado para monitorar um determinado processo foi:

strace –t –p (numero do processo) –o (arquivo)

Onde:

-*t*: Serve para imprimir a hora para cada linha de saída do *strace*;

-*p*: Opção para executar o *strace* utilizando o ID do processo, onde poderá será identificado utilizando o comando *ps ax* descrito na figura 4.2;

-*o*: Guarda o resultado obtido em um arquivo;

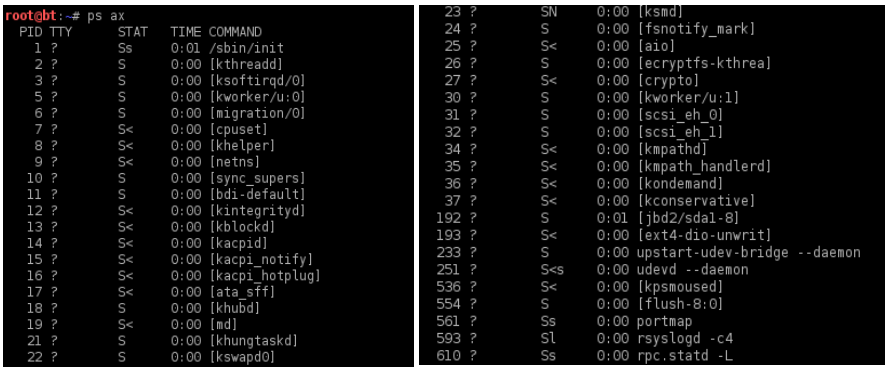

Figura 4.2: Trecho do comando ps ax em execução

Após coletadas as informações necessárias. O analisador, ou IDS, irá consultar em uma base de dados de ataque, informações necessárias que se caracterizam a existência de um ataque, analisando os parâmetros das funções das chamadas ao sistema coletadas, quantidade de vezes que uma chamada ao sistema realiza por segundo, dentre outras.

Segundo de Paula (2004), se refere a uma detecção baseado em conhecimento, devido o banco consistir apenas da coleta de informações relacionadas a outros ataques já conhecidos, verificando assim, ataques semelhantes.

Após a análise do processo, deve ser informado se existe ou não a suspeita de ataque. Inicialmente, não se pretende evitar a execução do ataque devido o mais importante ser apenas informar ao usuário a existência do mesmo.

5 – **Resultados e Discussão**

Exploits utilizados para análise: T50, *Slowloris*, *Joomscan*, *Skipfish* e *Plecost*.

A seguir são mostrados trechos dos arquivos gerados pelo comando *strace* demonstrado anteriormente que será analisado pelo programa:

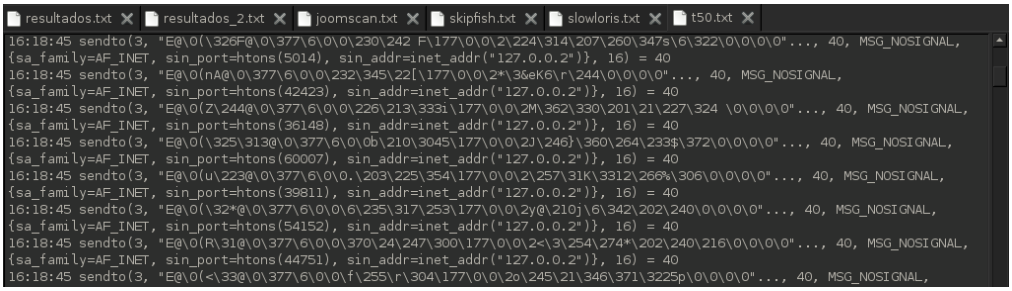

Figura 04: Arquivo gerado pelo strace monitorando o PID do *exploit* T50

O programa verifica a quantidade de chamadas que uma determinada função realiza por segundo e salva em um arquivo chamado resultados_2.txt. A imagem abaixo mostra o resultado obtido pelo programa após a análise do PID do *exploit* T50 em execução.

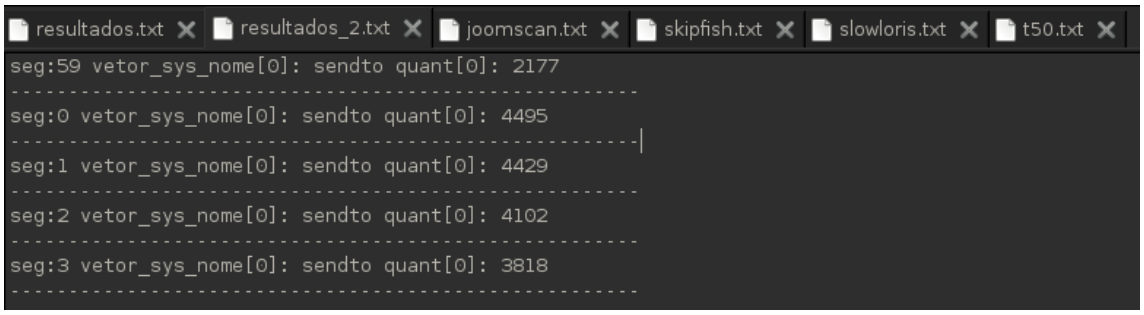

Figura 05: Arquivo gerado pelo código monitorando o PID do *exploit* T50

Pode-se observar que o comando *sendto* foi a única chamada ao sistema a ser executada pelo *exploit*. O comando *sendto* envia uma mensagem a um destino específico, ou seja, envia diversos pacotes a fim de ocasionar a indisponibilidade de um servidor ou site.

6 - **Conclusões**

Diversas alternativas de segurança são entregues a sociedade visando promover soluções para escapar das demais ameaças existentes a Internet, devido à sua excessiva utilização, que tem se tornado, de fato, um hábito na vida das pessoas. Sendo assim, foi modelo um código, com o intuito de auxiliar em trabalho futuros na detecção de ataques ao sistema.

7 – **Agradecimentos**

Agradeço o apoio de minha família, do orientador, do coordenador, dos colegas acadêmicos, da UEMS (Universidade Estadual de Mato Grosso do Sul) pelo auxílio pecuniário, e a todos que, diretamente ou indiretamente, contribuíram para a realização desse trabalho.

8 – **Referências Bibliográficas**

NAKAMURA, E., de Geus, P. 2003. **Segurança de redes em ambientes cooperativos**. Editora Futura, São Paulo, 3ª edição.

NORTHCUTT, S., COOPER, M., FEARNOW, M., FREDERICK, K. 2001. ***Intrusion signatures and analysis***. New Riders Publishing, 1ª edição.

DE PAULA, F. 2004. **Uma arquitetura de segurança computacional inspirada no sistema imunológico**. Tese de doutorado, Instituto de Computação, Universidade Estadual de Campinas.

PETHIA, R. 2000. ***Internet security issues***. CERT Coordination Center. Disponível em:

<http://www.cert.org/congressional_testimony/Pethia_testimony25May00.html>.

ANDERSON, A. 2005. **Protótipo para auxiliar no controle de vulnerabilidades em serviços da Internet**. Trabalho de conclusão de curso. Unochapecó. Disponível em: <http://www.unochapeco.edu.br/saa/tese/2500/tcc2.doc>. (último acesso: 17 fev. 2012)

SILBERSCHATZ, A.; GALVIN, P.; GAGNE, G. 2004. **Fundamentos de sistemas operacionais**. LTC, 6ª edição.

TANENBAUM, ANDREW S. 2003. **Redes de Computadores**. Traduzido por Vanderberg D. de Souza. Rio de Janeiro: Elsevier. 767p. 4ª Edição

JUNIOR, P. R. 2010. **Scanner de segurança SKIPFISK do Google para sites**. Disponível em: <http://www.paulojr.info/artigos/Scanner%20de%20seguranca%20 SKIPFISHb%20do%20Google%20para%20sites.pdf>. (último acesso: 18 jun. 2012)

UCHÔA, J. Q. 2005. **Segurança Computacional**. Lavras: UFLA/FAEPE.

IGREJA, F. 2007. **Detector de Intrusão e Anomalias em Redes de Automação Industrial**. Disponível em: <http://www2.ele.ufes.br/~projgrad/documentos/PG2007_1/ filipeigreja.pdf>. (último acesso: 26 jun. 2012)

CARDOZO, H. A. M. **Rastreamento de processos com o *strace***. Disponível em: <http://ha-mc.org/node/14>. (último acesso: 26 jun. 2012)

LAUREANO, M., 2006. **Máquinas Virtuais e Emuladores***:* Conceitos, Técnicas e Aplicações. 1ª edição. São Paulo: Novatec Editora.

RENATO, P., 2012. ***Backtrack: Solução open source para pen test***. Disponível em: <http://www.slideshare.net/pseixas/backtrack-soluco-open-source-para-pen-test>.
(último acesso: 03 mai. 2012)

ALMEIDA, A. R., 2008. **Como funcionam os *exploits***. Disponível em: <http://www.invasao.com.br/2008/12/12/como-funcionam-os-exploits/>.(último acesso: 01 mai. 2012)