Sistema de Monitoração e Reconstrução de Atividades em Honeypots

Luiz Gustavo C. Barbato e Antonio Montes
Laboratório Associado de Computação e Matemática Aplicada
Instituto Nacional de Pesquisas Espaciais
CEP 12227-010 - São José dos Campos - SP
{lgbarbato,montes}@lac.inpe.br

Resumo

Utilizar máquinas preparadas para serem comprometidas (honeypots) visando o aprendizado das técnicas adotadas pelos invasores com os próprios invasores[20] é uma nova visão de segurança de sistemas que vem ajudando a reagir contra ataques e a mudar a mentalidade que segurança é sinônimo exclusivamente de proteção. Com base neste novo paradigma, este artigo apresentará um sistema desenvolvido com o objetivo de monitorar, de forma imperceptível, todas as atividades dos invasores nos honeypots e permitir um estudo detalhado de suas técnicas.

Abstract

The use of hosts prepared to be compromised (honeypots) providing information about the techniques used by the attackers, from the attackers themselves is a new vision of systems security that helps to react against attacks and change the mentality that systems security is closely associated with passive protection. Based in this new paradigm, this paper shows a system developed to monitor, in a imperceptible way, all the attackers' activities in the honeypots and permit a detailed study of their techniques.

1. Introdução

Desde muito tempo atrás, estratégias militares[20] de defesa bem sucedidas são aquelas criadas com base no conhecimento do inimigo, seus tipos de armas, seus métodos de ataques, suas táticas de guerra e, principalmente, o seu objetivo. Conhecendo suas armas é possível conduzir treinamentos práticos. Conhecendo seus métodos de ataque é possível desenvolver métodos de defesa. Conhecendo suas táticas de guerra é possível projetar mecanismos para combatê-lo. E conhecendo seus objetivos é possível entender a motivação que o incentivou a iniciar os ataques.

Assim como os militares, a área de segurança de sistemas de informação também pode utilizar estes conhecimentos análogos para se defender contra ataques. Esta busca motivou os profissionais de segurança a criarem metodologias para obter informações mais precisas sobre os atacantes[1].

[1]Neste artigo, os termos atacantes e invasores possuirão o mesmo significado. Serão interpretados como sendo indivíduos mal intencionados que atacam e/ou invadem sistemas.

Uma das metodologias criadas faz uso de ferramentas de pesquisa, conhecidas como *honeynets*, que permitem a captura e estudo de cada passo dos atacantes. *Honeynets* são redes preparadas para serem comprometidas, compostas por várias máquinas e equipamentos denominados *honeypots*. Em cada *honeypot* são instalados sistemas e aplicativos reais, com intuito de criar ambientes similares aos que são utilizados em muitas organizações atualmente. Além dos sistemas e aplicativos, outros recursos são necessários para monitorar as operações efetuadas pelos atacantes durante um processo de invasão.

Um dos maiores problemas deste processo é capturar estas atividades sem que os atacantes percebam que estão sendo monitorados. Esta monitoração não é um processo trivial, pois envolve várias situações delicadas que requerem certos cuidados para não comprometerem todo o processo de pesquisa.

Nas primeiras implementações de *honeynets*[18], este tipo de monitoração era feito diretamente pelos sistemas de detecção de intrusão, visto que estes estão implantados na rede em pontos estratégicos para capturar e analisar todo o tráfego. Com essa estrutura, é possível remontar, por exemplo, todos os comandos executados e respostas em uma transação FTP ou uma sessão telnet ou ainda em uma conversa de IRC.

O sistema de detecção de intrusão oferece esta facilidade porque essas informações trafegam em claro pela rede, ou seja, informações não criptografadas. No entanto, esta topologia apresenta problemas com relação à conexões criptografadas, como uma sessão SSH por exemplo, pois para analisar o conteúdo dos pacotes seria necessário um poder computacional muito grande para decifrá-los. Neste caso a monitoração do tráfego de rede através de sensores não funciona, uma vez que a comunicação é criptografada. A maioria dos *backdoors* já possui a característica de sessão segura, permitindo a monitoração do tráfego somente até a sua instalação[5].

Com base nestas dificuldades, o presente trabalho apresenta um sistema desenvolvido com a intenção de capturar, de forma imperceptível, todas essas atividades, tanto em transações em texto puro quanto em transações criptografadas, visando a reconstrução das sessões das invasões.

2. Histórico

2.1. Honeypots

A primeira ocorrência publicada sobre monitoração dos passos dos atacantes ocorreu em 1988, quan-do Clifford Stoll[16] relatou um ataque sofrido pelo Lawrence Berkeley Laboratory(LBL). Um ano mais tarde, este relato se tornou um livro [17]. Nesta invasão sofrida pelo LBL, ao invés de tentarem bloquear os intrusos, Stoll e sua equipe resolveram permitir o acesso ao sistema enquanto monitoravam todas as atividades dos atacantes até que suas origens fossem descobertas, o que levou quase um ano. Para acompanhar os passos dos atacantes, impressoras foram instaladas em todas as portas de entrada do sistema, configuradas para imprimir todas as atividades que os atacantes realizavam nas máquinas. Com a ajuda dessas informações, foi possível identificar não só a origem dos invasores, mas também quais eram seus objetivos e motivações, quais redes foram comprometidas e quais redes os invasores estavam interessados em atacar.

Em 1992, Bill Cheswick[3] publicou um artigo descrevendo o acompanhamento de uma invasão ocorrida em 7 de janeiro de 1991 no Laboratório Bell da AT&T. Esta invasão ocorreu no *gateway* do laboratório que estava preparado para ser comprometido, rodando várias aplicações falsas tais como: FTP, telnet, SMTP, finger e rlogin. O intuito desta monitoração era descobrir quem estava interessado em atacar o laboratório; de onde vinham e qual a frequência destes ataques; e em que tipos de vulnerabilidades os atacantes estariam mais interessados. Esta monitoração durou meses, e contou com a ajuda de várias pessoas como Steven M. Bellovin [2] que desenvolveu várias ferramentas utilizadas para atrair, conter e capturar as ações dos invasores.

Em 2002, Lance Spitzner definiu claramente o conceito de *honeypot* como sendo "*Um recurso de segurança preparado para ser sondado, atacado ou comprometido*", e explicou a sua real função "*Honeypots não são uma solução, eles não consertam nada, são apenas uma ferramenta. E o seu uso depende do que se deseja alcançar*".

Em 2003, Martin Roesch[12], o criador do Snort, categorizou o uso dos *honeypots* em dois tipos: produção ou pesquisa. Para Roesch, *honeypots de produção são utilizados para diminuir os riscos e ajudar a proteger as redes das organizações; e os honeypots de pesquisa são designados para estudar e obter informações da comunidade dos atacantes*.

Os *honeypots* também podem ser classificados co-mo de baixa(serviços falsos) ou alta interatividade (serviços reais), nos quais os atacantes podem obter acesso total ao sistema. Os *honeypots* de alta interatividade, foco do artigo, não emulam serviços, ao invés disso, são instalados sistemas operacionais e aplicativos reais para os invasores interagirem com a máquina e permitir que seus passos sejam acompanhados.

2.2. Honeynets

A idéia de se criar uma rede preparada para ser comprometida visando o aprendizado, começou com uma pequena iniciativa de um analista de segurança da Sun Microsystems, chamado Lance Spitzner[18], que resolveu conectar uma máquina Linux Red Hat 5.0 à Internet em 25 de fevereiro de 1999. Sua intenção era fazer com que a comunidade dos atacantes lhe mostrasse como agia para rastrear, atacar e explorar sistemas.

Surpreendentemente, sua armadilha funcionou em 15 minutos, quando sua máquina Linux foi invadida com sucesso. Infelizmente, o atacante percebeu que estava sendo monitorado e apagou todo o conteúdo do sistema, destruindo todos os *logs* e ferramentas utilizadas durante o ataque.

Em consequência disso, Spitzner resolveu preparar todo um ambiente para capturar os dados, de for-ma que os atacantes não conseguissem mais apagá-los.

Com a evolução desta idéia surgiu em 25 de abril de 1999 o *Honeynet Project*[2], que inicialmente foi formado por 30 profissionais interessados em aprender sobre as ferramentas, táticas e motivações dos atacantes, e além disso compartilhar estas informações visando o aumento da segurança de sistemas.

Só em junho de 2000 o projeto entrou em operação efetiva, quando um dos *honeypots* foi invadido. A invasão ocorreu em um Solaris 2.6 por um grupo organizado de atacantes, tendo sido necessário contar com as habilidades de todos os integrantes do projeto para capturar os dados do ataque.

Esta experiência motivou o grupo a desenvolver uma ferramenta denominada de *honeynet*, que nada mais é que uma rede preparada para ser comprometida, ou seja, uma espécie de laboratório que fica exposto à Internet para ser atacado, porém com alguns mecanismos de monitoração, coleta e contenção de dados.

Com o passar do tempo, foram definidos dois elementos cruciais para a criação e manutenção bem sucedida de uma *honeynet*[18]. São eles: contenção e captura de dados. A contenção de dados foi definida como sendo uma forma de impedir que os atacantes, após invadirem a *honeynet*, a utilizem para atacar outras redes e a captura de dados foi definida com sendo uma maneira de coletar o maior número possível de informações, com o objetivo de promover um estudo detalhado dos passos dos atacantes.

Com a finalidade de juntar várias instituições internacionais envolvidas com pesquisas de *honeynets* foi criada a *Honeynet Research Alliance*[3]. Um dos projetos que faz parte do *Honeynet Research Alliance* é o Honeynet.BR[4][5], cri-

[2] http://www.honeynet.org
[3] http://www.honeynet.org/alliance
[4] http://www.honeynet.org.br

ado em dezembro de 2001 no INPE com intuito de acompanhar as atividades maliciosas na Internet brasileira.

3. Descrição do sistema

SMaRT (Session Monitoring and Replay Tool) é um sistema desenvolvido com o intuito de capturar todas as atividades de interação entre o atacante e o *honeypot*, ou seja, registrar todas as informações que passam no terminal do atacante, incluindo teclas pressionadas, comandos executados e seus retornos, visualização de textos, senhas de *backdoors*, etc.

Com essas informações capturadas é possível reconstruir as atividades do ataque em uma outra máquina, para analisar de forma detalhada o que o atacante fez durante a invasão. Uma outra possibilidade que está sendo estudada é acompanhar os ataques à medida que vão ocorrendo[17].

Os resultados do SMaRT podem ser comparados aos do aplicativo *script*, porém com as vantagens de monitoração em *kernel space*[1][11].

3.1. Arquitetura do sistema

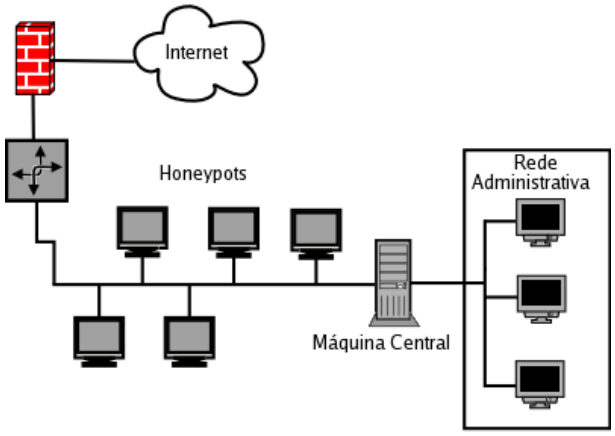

Figura 1: Arquitetura do sistema

Em cada *honeypot* é instalado o módulo *smartl* (*smart logger*), que fica escondido na máquina capturando todos os eventos de interatividade entre o invasor e *honeypot*. O próprio *smartl* envia estes dados para a rede por meio de pacotes UDP criptografados[1]. A emissão desses dados é transparente para o invasor, mesmo que este utilize monitoradores de tráfego de rede como o *tcpdump*.

No mesmo barramento existe uma máquina denominada central que trabalha em modo promíscuo, ou seja, analisa tudo que passa pela rede[6][7]. Estes dados são coletados, processados e armazenados em arquivos texto na própria máquina, através do módulo *smartc* (*smart collector*).

Além da interface de rede que coleta estes dados, a máquina central também possui uma outra interface conectada a uma rede denominada administrativa. Nesta, roda um outro módulo do sistema, nomeado de *smarts* (*smart server*), cuja função é disponibilizar os dados para os administradores da *honeynet* através de uma aplicação servidora TCP[15].

Quando os administradores da *honeynet* pretenderem analisar os dados, eles poderão se logar na máquina central ou utilizar um outro módulo do sistema que roda em máquinas da rede administrativa, o *smartg* (*smart gui*), que é uma aplicação cliente TCP[15] com uma interface desenvolvida em *ncurses* (Fig. 2).

3.2. Módulos do sistema

O sistema foi dividido em quatro módulos de acordo com a arquitetura apresentada, onde somente o primeiro módulo é dependente do sistema operacional, os demais podem ser compilados tanto em sistemas Linux quanto FreeBSD ou OpenBSD.

3.2.1. Smartl

O *smartl* é um módulo de *kernel* (LKM)[4], inicialmente desenvolvido para rodar em máquinas Linux, utilizado no registro e no envio dos dados capturados nos *honeypots*. Este é baseado na modificação do *sebek*[19], que já vem sendo utilizado em várias *honeynets* e apresentando bons resultados. Uma outra motivação em alterar o *sebek*[19] está relacionada com a técnica de esconder o tráfego de rede associado aos dados capturados.

Esta modificação consiste na troca do mecanismo de registro de atividades por um outro desenvolvido especialmente com o intuito de permitir a reconstrução e visualização de toda a sessão da invasão[1]. Antes do envio dos dados, os mesmos são submetidos a uma rotina simples de criptografia[1] apenas para esconder o conteúdo dos pacotes de outras máquinas da *honeynet* que não possuem este módulo instalado.

Para esconder o módulo após o seu carregamento, foi utilizado o módulo *cleaner* que acompanha o *rootkit*[10] *adore*. Este utiliza a técnica de remover o último módulo carregado da lista de módulos mantida em memória pelo *kernel*[8]. Assim é possível iludir comandos como o *lsmod* e arquivos como */proc/modules*. Apenas lembrando que esta técnica não esconde por completo o módulo do sistema, existem outras técnicas que utilizam métodos de força bruta para detectar módulos escondidos[13].

O novo registro de atividades inserido no *sebek*[19] utiliza a técnica de interceptar as chamadas de sistema[14]: *sys_read* e *sys_write*; e interceptar também as funções de terminal: *tty_read* e *tty_write*[1]. A utilização destas técnicas depende do seguinte algoritmo:

***Se** (existe um terminal associado ao processo)*

> ***Então:** Registre todas as operações de leitura e escrita no terminal.*
> ***Senão:** Registre todas as atividades de leitura e escrita no sistema referentes ao que é digitado e visualizado pelo invasor.*

Fim-Se

É importante destacar que não são capturados informações de rede que mostrem o local de onde os ataques estão partindo. Para isso, outros mecanismos são utilizados em uma *honeynet*, os quais não serão abordados neste artigo.

A instalação do *smartl* no *honeypot* continua sendo feita da mesma forma que o *sebek*[19], precisando que os seguintes parâmetros sejam passados ao carregar o módulo:

- *destination_ip*: Endereço IP de destino dos pacotes
- *destination_mac*: Endereço físico de destino dos pacotes
- *destination_port*: Porta de destino dos pacotes
- *interface*: Interface de saída dos pacotes

Estes parâmetros também podem ser facilmente configurados no *script* de instalação do *sebek*[19], *sbk_install.sh* que acompanha o *smartl*.

3.2.2. Smartc

O *smartc* é uma aplicação criada para capturar os dados exportados pelos *honeypots*. Este módulo trabalha como monitorador de tráfego de rede, desenvolvido com a biblioteca *pcap*, e fica constantemente analisando tudo que passa pelo barramento, retirando somente os dados enviados pelo *smartl*. A decisão pela utilização desta biblioteca foi por motivos de portabilidade para outros sistemas operacionais, além de permitir a seleção dos pacotes de uma forma bem simples e rápida com a utilização de filtros BPF[9].

Logo após a captura, os dados são decifrados e repassados para uma rotina que retira caracteres indesejados, como os referentes a cores nos terminais, caracteres de controle de certos editores de texto e outros utilizados na formatação de textos.

Depois deste procedimento, estes dados são armazenados em um arquivo texto, assim como outros criados para registrar informaçãoes de PID, GID, UID, terminal (caso haja) referentes ao processo que gerou os dados no *honeypot*.

O arquivo de sessão contém um caractere de controle no início de cada linha, indicando se a operação é de entrada (1 - IN) ou de saída (0 - OUT). Assim, é possível filtrar, por exemplo, somente os comandos digitados utilizando uma simples expressão regular que retire somente as linhas que começam com o número 1.

O *smartc* também foi projetado para trabalhar com dados *offline*. Ele pode ser utilizado para analisar dados capturados por outros analisadores de tráfego de rede como, por exemplo, o *tcpdump* e o *ethereal* que conseguem armazenar os dados em arquivos no formato do *tcpdump*. Com isso é possível utilizar o próprio *tcpdump* para capturar os dados, repassando estes arquivos para o *smartc* retirar e remontar os dados de sessão.

3.2.3. Smarts

O *smarts* é uma aplicação servidora[15] simples, que utiliza o protocolo TCP para transporte dos dados. Este módulo foi desenvolvido para disponibilizar os dados das sessões para os administradores da *honeynet* e, inicialmente atende a três tipos de comandos pré-definidos:

1. Obter lista de *honeypots*
   Este comando retorna a lista de todos os *honeypots* que foram comprometidos.
2. Obter lista de sessões de um *honeypots*
   Este comando retorna a lista dos horários de início das sessões de um *honeypot* específico.
3. Obter sessão
   Este comando retorna o conteúdo da sessão escolhida no item anterior

3.2.4. Smartg

A função deste módulo é criar uma interface de interação com o sistema, de forma que os administradores da *honeynet* possam selecionar e acompanhar as atividades nos *honeypots*. Este atua como uma aplicação cliente[15] simples TCP que se comunica com o *smarts* para receber os dados das sessões dos atacantes.

A interface do sistema foi implementada utilizando a biblioteca *ncurses* (Fig.2) visando a apresentação dos dados de uma forma amigável e, principalmente, eliminando a dependência do sistema de janelamento *X*. A utilização desta biblioteca também facilita a portabilidade para outros sistemas UNIX.

3.3. Precauções

Uma precaução é com relação a configuração da máquina onde rodarão estes módulos. Como mostrado na arquitetura da solução, esta máquina possui duas placas de rede, uma ligada a *honeynet* e a outra a rede administrativa. Cada módulo trabalha em uma interface diferente, o *smartc* na interface da *honeynet* e o *smarts* na interface da

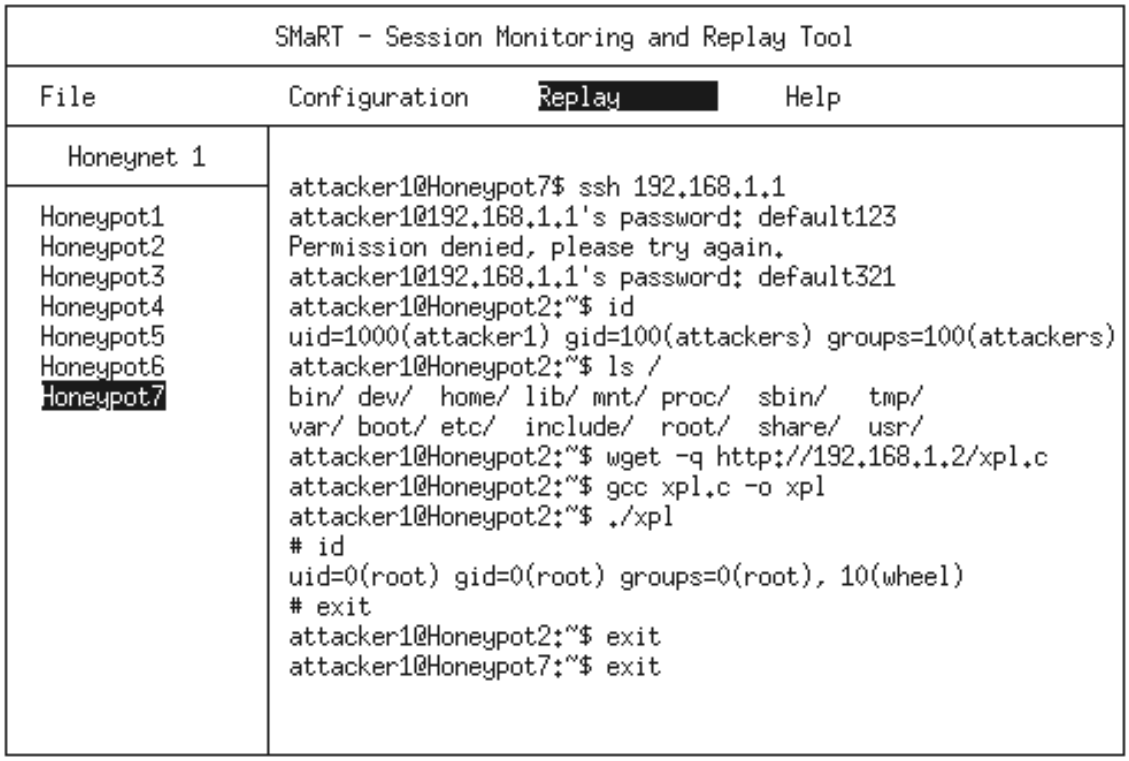

Figura 2: Captura de tela do SMaRT reconstruindo uma sessão

rede administrativa. A máquina central deve ser bem configurada para não permitir que os atacantes acessem a rede administrativa por meio de alguma falha de segurança que este sistema possa apresentar.

Para minimizar os riscos de segurança possíveis no sistema, os módulos *smartc* e *smartl* rodam em um ambiente enjaulado e com baixos privilégios. Porém, para que estas funcionalidades sejam habilitadas é necessário que ambos sejam inicializados pelo super usuário e o sistema automaticamente se encarregará de fazer as modificações depois que certas operações forem executadas.

Uma outra precaução é utilizar criptografia nas transações tanto entre o *smartl* e o *smartc* quanto entre o *smarts* e o *smartg* para ocultar realmente as informações de pessoas não autorizadas, mantendo assim, a confidencialidade das mesmas.

Está sendo adotado no SMaRT certos cuidados com relação a política de desenvolvimento para não permitir que o sistema seja utilizado para outras finalidades hostis. Dentre esta política adotada pode-se dizer que: os pacotes não possuem o IP de origem alterado e as portas de origem e destino são as mesmas, bem como os endereços físicos de origem e destino. Essas medidas facilitam aos administradores de rede identificarem, através de filtros BPF[9], a utilização ilícita deste sistema em suas redes.

4. Conclusões

Conhecer quem são os nossos inimigos para projetar mecanismos de defesa é uma atividade antiga utilizada em estratégias militares[20]. Os profissionais de segurança de sistemas também utilizam destes conhecimentos para se defenderem contra ataques.

Para buscar esses conhecimentos é necessário utilizar ferramentas de pesquisa como *honeypots* e *honeynets*, com a ajuda de outros mecanismos de coleta de dados. O SMaRT (Session Monitoring and Replay Tool), apresentado neste artigo, está sendo desenvolvido para auxiliar na coleta desses conhecimentos, proporcionando um estudo detalhado das atividades nas máquinas comprometidas.

Como trabalho futuro espera-se terminar a funcionalidade de acompanhar os ataques à medida que eles ocorrem e, assim, permitir a visualização imediata da invasão.

Para melhorar a operação do *smartl* almeja-se desenvolver um outro módulo para detectar alterações na tabela de chamadas de sistema e permitir que *rootkits* de *kernel*[10] sejam instalados pelos invasores sem cancelar o sistema de monitoração nem atrapalhar o funcionamento dos mesmos.

Será adicionado ao *smartl* rotinas para esconder a real localização do módulo em disco e o local onde o mesmo é carregado, dificultando para o atacante perceber evidências que possam induzí-lo a desconfiar da máquina comprometida.

Pretende-se também extender a reconstrução de sessão para outros sistemas operacionais, de forma que todos os *honeypots* da *honeynet* possam ser monitorados.

Várias outras idéias poderão surgir para melhorar o SMaRT. Este é um desenvolvimento que pretende ser contínuo para contribuir nos estudos e pesquisas de segurança dos sistemas.

Agradecimentos

Os autores agradecem a ajuda do grupo Honeynet.BR[5] pelas contribuições no desenvolvimento do projeto SMaRT e a Adriene Emergente pela revisão deste artigo.

Referências

[1] Luiz Gustavo C. Barbato and Antonio Montes. Técnicas de Monitoração de Atividades em Honeypots de Alta Interatividade. a ser publicado nos Anais do V Simpósio sobre Segurança em Informática (SSI'2003).

[2] Steven M. Bellovin. There Be Dragons. In *Proceedings of the Third Usenix Security Symposium*, 1992. http://www.research.att.com/~smb/papers/dragon.ps (verificado em 20/10/2003).

[3] William R. Cheswick. An Evening with Berferd in Which a Cracker is Lured, Endured, and Studied. In *Proceedings of the Winter 1992 USENIX Conference*, pages 163–174, San Francisco, California, USA, 1992. http://cne.gmu.edu/modules/acmpkp/security/texts/CRACKER.PDF (verificado em 20/10/2003).

[4] Jonathan Corbet and Alessandro Rubini. *Linux Device Drivers*. O'Reilly & Assoc, 2001. ISBN 0-596-00008-1.

[5] Amândio Balcão Filho, Ana Sílvia M. S. Amaral, Antonio Montes, Cristine Hoepers, Klaus Steding-Jessen, Lucio Henrique Franco, and Marcelo H. P. Caetano Chaves. Honeynet.BR: Desenvolvimento e Implantação de um Sistema para Avaliação de Atividades Hostis na Internet Brasileira. In *Anais do IV Simpósio sobre Segurança em Informática (SSI'2002)*, pages 19–25, São José dos Campos, SP, Novembro 2002. http://www.honeynet.org.br/papers/hnbr-ssi2002.pdf (verificado em 20/10/2003).

[6] Gianluca Insolvibile. Kernel Korner: Inside the Linux Packet Filter. Linux Journal, Vol 94, Fevereiro, 1 2002. http://www.linuxjournal.com/article.php?sid=4852 (verificado em 20/10/2003).

[5] Amândio Balcão Filho, Antonio Montes, Cristine Hoepers, Klaus Steding-Jessen, Lucio Henrique Franco, Luiz Gustavo C. Barbato e Marcelo H. P. Caetano Chaves

[7] Gianluca Insolvibile. Kernel Korner: Inside the Linux Packet Filter II . Linux Journal, Vol 95, Março, 1 2002. http://www.linuxjournal.com/article.php?sid=5617 (verificado em 20/10/2003).

[8] Scott Andrew Maxwell. *Linux Core Kernel Commentary*. Coriolis, 2001. ISBN 1-588-80149-7.

[9] Steven McCanne and Van Jacobson. The BSD packet filter: A new architecture for user-level packet capture. In *USENIX Winter*, pages 259–270, 1993. http://citeseer.nj.nec.com/mccanne92bsd.html (verificado em 20/10/2003).

[10] Nelson Murilo and Klaus Steding-Jessen. Métodos para Detecção Local de Rootkits e Módulos de Kernel Maliciosos em Sistemas Unix. In *Anais do III Simpósio sobre Segurança em Informática (SSI'2001)*, pages 133–139, São José dos Campos, SP, Outubro 2001. http://www.chkrootkit.org/papers/chkrootkit-ssi2001.pdf (verificado em 20/10/2003).

[11] A. Silberschatz, G. Gagne, and P. Galvin. *Operating System Concepts*. Student Computing Series. John Wiley & Sons, 1999. ISBN 0-471-36508-4.

[12] Lance Spitzner. *Honeypots: Tracking Hackers*. Addison-Wesley, 2003. ISBN 0-321-10895-7.

[13] Phrack Staff. Linenoise. *Phrack Inc.*, 61(3), Agosto 2003. http://www.phrack.org/show.php?p=61\&a=3 (verificado em 20/10/2003).

[14] W. Richard Stevens. *Advanced Programming in the UNIX Environment*. Addison-Wesley Publishing Company, 1992. ISBN 0-201-56317-7.

[15] W. Richard Stevens. *UNIX Network Programming Volume 1 Networking APIs: Sockets and XTI*. Prentice-Hall, 2 edition, 1998. ISBN 0-13-490012-X.

[16] Clifford Stoll. Stalking the Wily Hacker. *Communications of the ACM*, 31(5):484–497, Maio 1988. http://cne.gmu.edu/modules/acmpkp/security/texts/HACKER.PDF (verificado em 20/10/2003).

[17] Clifford Stoll. *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*. Doubleday, Garden City, NY, 1989. ISBN 0-385-24946-2.

[18] The Honeynet Project. *Know Your Enemy: Revealing the Security Tools, Tactics, and Motives of the Blackhat Community*. Addison-Wesley, 1st edition, Agosto 2001. ISBN 0-201-74613-1.

[19] The Honeynet Project. Know Your Enemy: Sebek2 A kernel based data capture tool. Setembro 2003. http://www.honeynet.org/papers/sebek.pdf (verificado em 20/10/2003).

[20] Sun Tzu. *The Art of War*. Random House, 1981. ISBN 0-877-73452-6.