# O Projeto Honeynet.BR

Honeynet.BR Team*
*Laboratório Associado de Computação e Matemática Aplicada*
*Instituto Nacional de Pesquisas Espaciais*
honeynet-team@lac.inpe.br

## Abstract

*A honeynet is a research tool consisting of a network specifically designed for the purpose of being compromised. Once compromised, the honeynet can be used to observe the intruders' activities, tactics, tools and motives. In this paper we discuss the concepts involved with this research area and present the Honeynet.BR Project, its implementation, developed tools and results.*

## Resumo

*Honeynets são ferramentas de pesquisa que consistem de uma rede projetada especificamente para ser comprometida. Uma vez comprometida, a honeynet é utilizada para observar o comportamento dos invasores, suas táticas, ferramentas e motivações. Neste artigo serão introduzidos os conceitos desta área de pesquisa e apresentado o Projeto Honeynet.BR, sua implantação, ferramentas desenvolvidas e resultados obtidos.*

## 1. Introdução

Nos últimos anos tem crescido a necessidade da comunidade de segurança de entender os ataques e o perfil dos atacantes de redes conectadas à Internet. Com este intuito alguns grupos[1] têm se dedicado a desenvolver métodos que permitam detectar e acompanhar ataques a redes de computadores. Um dos métodos que tem sido utilizado é o desenvolvimento, implantação e monitoração de *honeynets*.

*Honeynets* são ferramentas de pesquisa que consistem de uma rede projetada especificamente para ser comprometida [11, 7]. Uma vez comprometida, a *honeynet* é utilizada para observar o comportamento dos invasores, possibilitando a realização de análises detalhadas das ferramentas utilizadas, de suas motivações e das vulnerabilidades exploradas.

Neste artigo serão apresentados inicialmente os conceitos e histórico de *honeypots* e *honeynets*, seguidos da descrição do Projeto Honeynet.BR e dos detalhes de sua implantação. São discutidos a seguir os métodos e ferramentas desenvolvidas para contenção de tráfego de saída, captura de tráfego e geração de alertas e sumários. Por fim, serão apresentados os resultados obtidos e sugestões de trabalhos futuros.

## 2. Conceitos e Histórico

As primeiras referências à implementação de mecanismos de acompanhamento das atividades de invasores datam de 1988, quando Clifford Stoll [10, 9] tornou pública a história da invasão ocorrida nos sistemas do Lawrence Berkeley Laboratory (LBL), e 1992 quando Bill Cheswick [2] e Steven Bellovin [1] publicaram artigos descrevendo o acompanhamento de uma invasão em um dos sistemas da AT&T.

Em 1998 Fred Cohen desenvolveu o *Deception Toolkit* (DTK) [3], uma ferramenta cujo objetivo é explicitamente iludir atacantes. Nesta época surgiu o termo *honeypot* como definição para um recurso de segurança preparado especificamente para ser sondado, atacado ou comprometido e para registrar essas atividades [8].

Em 1999 teve início o *Honeynet Project* [11, 7], onde um grupo de pesquisadores e profissionais da área de segurança iniciaram a criação de uma rede especificamente projetada para ser comprometida. O objetivo do projeto é revelar as ferramentas, táticas e motivações dos invasores.

A partir deste projeto foi criada a *Honeynet Research Alliance*[2], que é uma proposta de trabalho conjunto de diversas instituições internacionais envolvidas com pesquisa na área de *honeynets*.

## 3. O Projeto Honeynet.BR

A Honeynet.BR entrou em operação em março de 2002, porém a sua fase de projeto teve início em dezembro de 2001. Nesta fase foram tomadas as principais decisões referentes às suas características, tais como topologia, sistemas operacionais e ferramentas a serem adotadas, além do início do desenvolvimento de ferramentas próprias para análise e contenção de tráfego.

A topologia da Honeynet.BR, exibida na Fig. 1, está dividida em duas partes distintas: a Rede Administrativa e a

*Amândio Balcão Filho, Ana Sílvia M. S. Amaral, Antonio Montes, Cristine Hoepers, Klaus Steding-Jessen, Lucio Henrique Franco e Marcelo H. P. Caetano Chaves.

[1] http://www.honeynet.org/alliance/

[2] idem.

*Honeynet* propriamente dita.

A Rede Administrativa tem as funções principais de conter a saída de tráfego malicioso da *Honeynet* e monitorar todo o tráfego, seja ele interno ou não. Ela é transparente tanto para a Internet quanto para a própria *Honeynet*. Esta rede é composta por:

- Um Firewall, que permite a entrada de todo o tráfego para a *Honeynet*, mas possui regras para impedir a saída de tráfego malicioso (as funcionalidades do Firewall serão discutidas em detalhes na seção 4);
- Uma máquina (Hogwash), configurada de modo a bloquear a saída de tráfego com conteúdo sabidamente malicioso (detalhes na seção 4.3);
- Um IDS (*Intrusion Detection System*), que captura e analisa o tráfego da *Honeynet* e emite alertas no caso de seu comprometimento. Também é responsável pela emissão de sumários diários sobre a atividade observada (detalhes da implementação nas seções 5 e 6);
- Uma máquina destinada a armazenar artefatos[3] e imagens dos discos dos *hosts* da Honeynet.BR quando de seu comprometimento (Forensics).

Diferentemente de outras *honeynets* que se tem conhecimento, os principais mecanismos de contenção e geração de alertas da Honeynet.BR foram desenvolvidos por membros do projeto, tendo o sistema operacional OpenBSD como sua plataforma principal.

A *Honeynet* é composta por diversos *hosts*, que são *honeypots* com sistemas operacionais e arquiteturas variadas, de modo a permitir que seja possível observar o comportamento de invasores em diversas plataformas. Um destes *hosts* opera como servidor de nomes para a *Honeynet*, além de possuir o serviço de `syslog` habilitado, atuando como servidor central de *logs* para os demais *hosts*. Na próxima seção serão discutidos detalhes do processo de instalação e acompanhamento dos *honeypots*.

## 4. Contenção de Tráfego de Saída

Um dos pré-requisitos mais importantes da *Honeynet* é a contenção de tráfego malicioso de saída, pois o seu objetivo é acompanhar as ações dos invasores, e não prover meios para a deflagração de ataques.

Tipicamente, após o comprometimento de um sistema, o invasor inicia *scans*, ataques de *Denial of Service* ou tentativas de comprometer outras redes. O desafio da *Honeynet* é conter esse tipo de atividade maliciosa sem, contudo, inibir outras atividades de interesse, como por exemplo *download* de ferramentas ou comunicação com outros invasores.

[3]Artefatos podem ser definidos como todo o material deixado pelo invasor após o comprometimento de uma máquina.

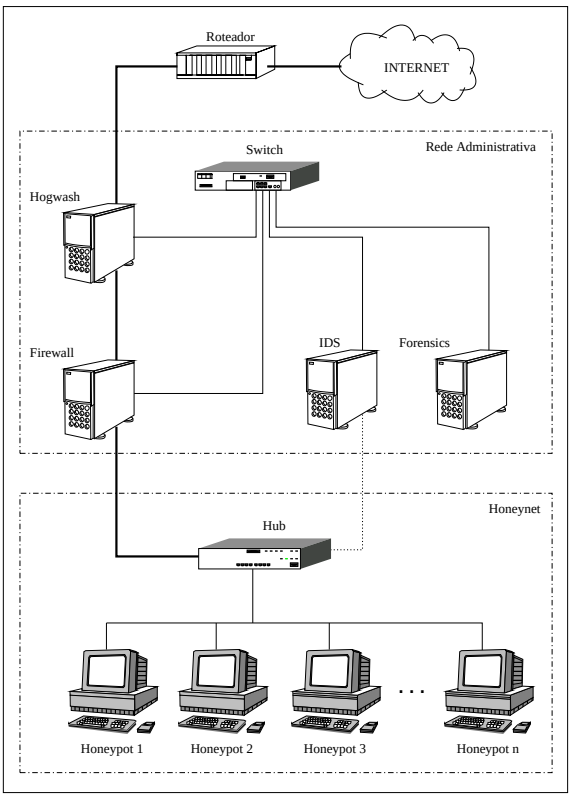


**Figura 1: Topologia da Honeynet.BR.**

A seguir são descritos os métodos implantados na Honeynet.BR para contenção de tráfego de saída.

### 4.1. Regras de Saída do *Firewall*

O Firewall está configurado para atuar como uma *bridge*: não possui endereço IP nas suas interfaces e não decrementa o TTL (*Time to Live*) dos pacotes IP que o atravessam. Desse modo as chances do Firewall ser detectado e atacado são menores.

O filtro de pacotes `pf` (*OpenBSD Stateful Packet Filter*) [4] é utilizado para implementar as regras que permitem qualquer tipo de tráfego de entrada, mas descartam tráfego potencialmente malicioso de saída da *Honeynet*.

### 4.2. Alteração Dinâmica de Regras do *Firewall*

O filtro de pacotes `pf`, utilizado no Firewall, é um filtro de pacotes *stateful*, de modo que não inspeciona apenas pacotes individuais, mas utiliza o conceito de sessões estabelecidas [4].

O Firewall está configurado para, caso um pacote não seja

descartado por nenhuma regra da seção 4.1, criar entradas na tabela de estados para pacotes saindo da *Honeynet*.

Inspecionando-se, num dado momento, a tabela de estados do `pf` é possível determinar várias informações sobre as sessões em andamento.

Foi desenvolvida uma ferramenta de código aberto, denominada `sessionlimit`, para monitorar continuamente as entradas da tabela de estados e interagir com o `pf`, inserindo e retirando regras de filtragem conforme necessário.

O `sessionlimit`, monitorando as sessões de saída, pode bloquear o tráfego de um *host* da *Honeynet* usando um dos critérios abaixo:

1. Taxa de crescimento muito rápida no número de sessões associadas a um IP de origem[4];
2. Número máximo de sessões associadas a um IP de origem;
3. Número máximo de *bytes* de uma sessão ICMP.

Caso uma dessas condições seja satisfeita, uma regra bloqueando o tráfego deste *host* é inserida nas regras correntes do `pf` e as sessões de saída deste IP, em andamento, são removidas.

É importante notar que a regra de bloqueio inserida pelo `sessionlimit` afeta apenas o tráfego de saída. As sessões já estabelecidas de entrada para este *host*, que tipicamente incluem a sessão interativa do invasor, não são afetadas.

Uma regra de bloqueio expira após um certo tempo[5], sendo então retirada pelo `sessionlimit` da lista de regras ativas do Firewall.

### 4.3. Bloqueio de Saída por Conteúdo

Além dos métodos já descritos, e que são implementados no Firewall, pacotes maliciosos saindo da *Honeynet* podem ser descartados pela máquina Hogwash, dependendo do resultado da análise do seu conteúdo.

Esta análise é feita utilizando-se a ferramenta de código aberto `hogwash`[6], que permite descartar pacotes cujo conteúdo possua a assinatura de um ataque conhecido.

Esta ferramenta utiliza as mesmas regras da ferramenta de detecção de intrusão `snort` [6] e executa em uma máquina dedicada para este fim, como mostrado na Fig. 1.

Uma vantagem da utilização desta ferramenta é a fácil atualização das assinaturas, que podem vir tanto da comunidade de segurança em geral como podem ser localmente criadas em função de tráfego malicioso anteriormente observado na *Honeynet*.

[4] Esta taxa é configurável.

[5] Atualmente 30 minutos, mas esse valor é configurável.

[6] `http://sourceforge.net/projects/hogwash/`

### 4.4. Limitação de Banda

Decidiu-se, como uma medida adicional, limitar a banda disponível de saída da *Honeynet* através do uso de `ALTQ` (*Alternate Queueing*).[7]

A intenção é limitar a intensidade de um ataque de *Denial of Service* caso os demais mecanismos de contenção de tráfego falhem. Desse modo o invasor não terá a sua disposição toda a banda disponível, mas apenas uma parte desta.

## 5. Captura de Tráfego

Todo tráfego que entra e sai da *Honeynet*, bem como o tráfego interno entre os *hosts*, é armazenado.

A captura de tráfego é feita em dois pontos:

1. Firewall

   Todo o tráfego que entra e sai da *Honeynet* é registrado[8] no Firewall em formato `tcpdump` binário, através do mecanismo de *logging* do filtro de pacotes `pf` [4]. Este formato facilita a manipulação dos pacotes e de seu conteúdo, pois permite o uso de ferramentas populares como `tcpdump`, `ethereal`, `ngrep`, etc.

2. IDS

   O IDS possui uma interface sem endereço IP, que captura todo o tráfego entre os *honeypots* e saindo e entrando da *Honeynet*.

   O *script* de captura usa o `tcpdump` para a coleta dos dados, armazenando-os em arquivos referenciados pelo ano, mês, dia e horário do início da coleta.

   Os dados capturados pelo IDS também são utilizados pelos mecanismos de alerta e para geração de sumários diários de atividade, como descrito na seção 6.

### 5.1. Rotação e Compressão de Dados

Todos os dados capturados pelo Firewall e pelo IDS são rotacionados e comprimidos a cada 24 horas.[9] O nome do arquivo de captura contém o ano, mês e dia. Após 30 dias esses arquivos são movidos para um dispositivo de armazenamento *off-line*.

## 6. Geração de Alertas e Sumários

### 6.1. Alertas

A geração de alertas parte do princípio que qualquer tráfego observado na *Honeynet* é malicioso. Tráfego origina-

[7] `www.csl.sony.co.jp/person/kjc/software.html`

[8] Com exceção de pacotes com endereço de origem forjado, tipicamente usados em ataques de *Denial of Service*, devido ao grande volume de *logs* gerados.

[9] Este intervalo de tempo é configurável.

do de dentro da *Honeynet* para fora indica necessariamente uma máquina comprometida.

Os alertas podem ser gerados dos seguintes modos:

1. Tráfego de saída

   Um *script*, na máquina IDS, monitora o tráfego capturado. Qualquer pacote saindo da *Honeynet*, que não seja resposta a um pacote vindo de fora, gera um alerta. A saída é produzida com `tcpdump` e os alertas são agrupados e enviados periodicamente.

2. Comandos de *shell*

   As máquinas Unix da *Honeynet* possuem uma *shell* modificada que envia o histórico dos comandos via o serviço de `syslog` [11]. Um *script* executado na máquina IDS monitora o tráfego da rede e gera um alerta se detectar o padrão dos *logs* enviados pela *shell* modificada.

Um mesmo alerta pode conter os dois modos descritos acima. Uma cópia de todos os alertas é mantida na máquina IDS para referência futura.

Os alertas podem ser enviados por *email*, *pager* ou telefone celular.

### 6.2. Sumários

Diariamente é gerado um sumário da atividade registrada na *Honeynet* referente ao dia anterior. Esse sumário é enviado por *email* e também armazenado na máquina IDS.

Os dados de entrada são os arquivos comprimidos de captura do IDS, compreendendo um dia de captura de todo o tráfego da *Honeynet*. A saída contém:

1. Estatísticas

   São geradas estatísticas com o total de pacotes capturados, percentual de pacotes por protocolo e *hosts* que mais originaram tráfego.

2. Alertas de `snort`

   O programa `snort` [6] é usado para ler o arquivo de captura e gerar alertas, que são listados neste sumário.

3. Tráfego de entrada na *Honeynet*

   Saída do `tcpdump` mostrando tráfego destinado à *Honeynet* e originado de fora desta.

## 7. Resultados

Durante o período de observação da *Honeynet* foram detectadas inúmeras varreduras e diversas invasões que nos permitiram coletar ferramentas, acompanhar as vulnerabilidades mais exploradas e a troca de informações entre os invasores.

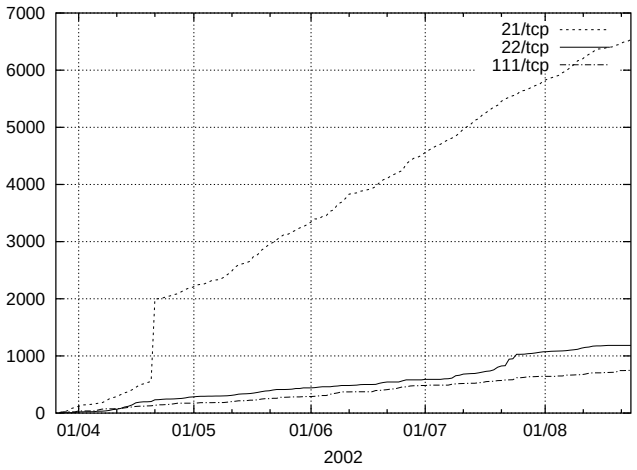


**Figura 2: *Scans* de `ftp`, `ssh` e `RPC`**

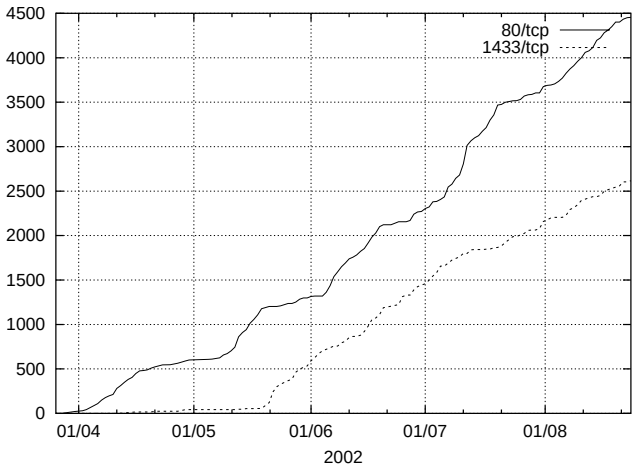


**Figura 3: Atividade de *worms***

Os *scans* mais freqüentes foram aqueles procurando por vulnerabilidades nos serviços de `ftp` (`21/tcp`), `ssh` (`22/tcp`) e serviços baseados em `RPC` (`111/tcp`). Os dados coletados, no período de 27 de março a 24 de agosto de 2002, referentes a estes *scans* podem ser observados na Fig. 2, que mostra seus valores acumulados.

Pudemos observar também um grande número de ataques realizados de maneira automatizada por *worms*. Estes ataques em geral eram destinados a servidores Web — inicialmente aos servidores IIS e posteriormente aos servidores Apache. Foi interessante notar que houve também uma grande atividade do *worm* que explora vulnerabilidades do Microsoft SQL Server. O aumento desta atividade coincidiu com a época de lançamento do *Incident Note*[10] do

[10] CERT *Incident Note* IN-2002-04
`http://www.cert.org/incident_notes/IN-2002-04.html`

**Tabela 1: Os 10 países que mais originaram *scans*, *exploits* e acessos a *backdoors*.**

| Origem dos *Scans* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| US | KR | BR | FR | CN | DE | TW | IT | RO | JP |
| 170 | 123 | 122 | 92 | 83 | 63 | 56 | 52 | 40 | 31 |
| Origem dos *Exploits* Lançados | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| US | KR | TW | CN | PL | JP | TH | IN | IT | BR |
| 24 | 11 | 8 | 8 | 5 | 5 | 4 | 4 | 3 | 3 |
| Acesso a *Backdoors* | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| RO | TW | BR | KR | CA | US | JP | CN | IL | PL |
| 54 | 14 | 13 | 5 | 4 | 4 | 4 | 2 | 1 | 1 |

CERT/CC sobre o assunto, em 22 de maio de 2002. A atividade de *worms* observada na *honeynet* durante o período de 27 de março a 24 de agosto de 2002 pode ser vista na Fig. 3. Esta figura mostra os valores acumulados relativos às tentativas de conexão dos *worms* para servidores Web (`80/tcp`) e SQL (`1433/tcp`).

Além disso, foi constatada uma grande procura por *proxies* abertos e servidores de *email* mal configurados, que permitissem sua utilização para envio de SPAM.

Os ataques que tiveram sucesso, em sua maioria, exploraram vulnerabilidades de `wu-ftpd`, `sshd` e `telnetd` a fim de obter acesso privilegiado aos *honeypots*. O perfil dos invasores era muito parecido: praticamente todos, após obter acesso privilegiado, instalavam ferramentas de *scan*, *exploits*, *massrooters*, *rootkits* e programas relacionados com IRC. Em alguns casos também foram instaladas ferramentas de *Denial of Service*.

Alguns *rootkits* novos, que não eram detectados pela ferramenta de código aberto `chkrootkit` [5], foram coletados e fornecidos aos seus autores para atualização da ferramenta.

Todos os *backdoors* instalados utilizavam algum mecanismo de criptografia, impossibilitando a captura do tráfego relativo a essas sessões. Desse modo, o acompanhamento das sessões interativas do invasor ficou restrito aos dados observados através da *shell* modificada.

A maior parte dos invasores disparou os ataques que comprometeram os *honeypots* a partir de máquinas do exterior, mesmo quando os invasores eram brasileiros.

Através das conversas observadas no tráfego de IRC capturado, foi possível verificar que alguns invasores eram brasileiros, mas que a grande maioria era composta por romenos.

Na Tab. 1 podemos ver com mais clareza a diferença entre as origens de *scans*, *exploits* lançados e acessos a *backdoors*. Pode-se ver que o acesso aos *backdoors* originou-se, na maioria das vezes, de IPs da Romênia e do Brasil, embora os *scans* e os *exploits* tenham sido lançados de máquinas situadas em outros países. Uma possível explicação é o grande número de máquinas comprometidas sendo usadas como base para *scans* e ataques. Nesta tabela os países estão identificados com o seu código ISO 3166 correspondente.

Os dados relativos às invasões de romenos foram resumidos e enviados para o *Honeynet Project*, que está elaborando o documento "*Know Your Enemy – A Profile: Romanian Blackhat Community*". Este perfil da comunidade romena de invasores está sendo traçado com dados obtidos de *honeynets* instaladas em diversos países. Este documento será publicado em breve.

## 8. Trabalhos Futuros

O uso cada vez mais disseminado de sessões criptografadas por parte dos invasores faz com que seja extremamente importante o desenvolvimento de outras técnicas para a monitoração de suas atividades. Planeja-se a implementação de ferramentas para a realização de captura de teclado (*keylogging*), que podem ser implementadas em módulos de *kernel* ou bibliotecas do sistema.

Além disso, com o objetivo de diminuir o tempo de resposta do `sessionlimit` e melhorar a performance do programa em redes de maior velocidade, planeja-se separar a sua funcionalidade em módulos de:

1. Detecção de *scans* e *Denial of Service* através da interação direta com a interface de rede;

2. Controle do número máximo de sessões estabelecidas, através da consulta da tabela de estados de saída.

Pretende-se também tornar o mecanismo de configuração da ferramenta mais sofisticado, com parâmetros distintos para cada *host* da *Honeynet*.

Outro projeto, que está em fase inicial de testes, é a utilização do sistema ACID (*The Analysis Console for Intrusion Databases*)[11], que é um sistema desenvolvido para processar e fazer buscas em uma base de dados de eventos de segurança gerados por diversas ferramentas de monitoração de redes.

## 9. Conclusões

Como primeira *honeynet* no Brasil dedicada à pesquisa e desenvolvimento de ferramentas que se tem conhecimento, o Projeto Honeynet.BR revelou-se de grande utilidade na coleta de artefatos e na avaliação de atividade hostil em redes brasileiras.

Foi possível observar também que a comunidade de invasores está usando exclusivamente ferramentas com criptografia para acesso às máquinas comprometidas, tornado

[11] `http://www.cert.org/kb/acid/`

inútil a captura de suas sessões de rede. Isto reforça a necessidade do desenvolvimento e uso de novos mecanismos de monitoração.

## 10. Agradecimentos

Várias pessoas e entidades ajudaram a viabilizar este projeto e gostaríamos de agradecer, em particular, ao Prof. Dr. Ulisses Thadeu Vieira Guedes, à Secretaria de Administração do Ministério de Ciência e Tecnologia e à FAPESP.

## Referências

[1] Steven M. Bellovin. There Be Dragons. In *Proceedings of the Third Usenix Security Symposium*, 1992.

[2] 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.

[3] Fred Cohen. Deception ToolKit. Risks Digest, Vol 19.62, March, 9 1998. `http://catless.ncl.ac.uk/Risks/19.62.html`.

[4] Daniel Hartmeier. Design and Performance of the OpenBSD Stateful Packet Filter (pf). In *Proceedings of the FREENIX Track: 2002 USENIX Annual Technical Conference (FREENIX '02)*, Monterey, California, USA, June 2002.

[5] 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.

[6] Martin Roesch. Snort — Lightweight Intrusion Detection for Networks. In *Proceedings of LISA '99: 13th Systems Administration Conference*, Seattle, Washington, USA, November 1999.

[7] Lance Spitzner. Learning the Tools and the Tactics of the Enemy with Honeynets. In *Proceedings of the 12th Annual Computer Security Incident Handling Conference*, Chicago, Illinois, USA, June 2000.

[8] Lance Spitzner and Marcus Ranum. Honeypots: Tracking Hackers. In *SANS 2002 Annual Conference*, Orlando, Florida, USA, April 2002.

[9] Clifford Stoll. Stalking the Wily Hacker. *Communications of the ACM*, 31(5):484–497, May 1988.

[10] 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.

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