Projeto e Implementação de um Network Operation Center baseado na Integração de Ferramentas Livres

Taciano B. de Oliveira[1], Henrique S. Pedroso[3], Luiz F. Zaltron[2], Gregório L. Ferreira[2], Jaziel Lobo[1], Sandra D. Piovesan[1], Roseclea D. Medina[1,2], Érico M. H. do Amaral [1,2,3]

[1] Universidade Luterana do Brasil (ULBRA)
Rua Martinho Lutero, s/n - CEP 96500-000, Cachoeira do Sul – RS - Brasil

[2] Universidade Federal de Santa Maria (PPGI/UFSM)
Av. Roraima, 1000, Cidade Universitária – CEP 97105-900 – RS - Brasil

[3] Centro Regional Sul de Pesquisas Espaciais (CRS/INPE-MCT)
Av. Roraima, S/N, Campus UFSM – CEP 97105-970 – RS - Brasil

{tacianobalardin, hsobrozapedroso, lfzaltron, greg.lofe, jaziel.lobo, sanpiovesan, roseclea.medina, ericohoffamaral}@gmail.com

Resumo. *Um NOC (Network Operation Center) é uma solução eficaz para o monitoramento e controle de ambientes computacionalmente conectados, por centralizar a gerência da rede em um ponto único, provendo um centro de programas que a monitoram, disponibilizando informações sobre cada ativo conectado. Este artigo apresenta um estudo sobre ferramentas para monitoramento e gerência de rede, além da implementação de um front end multiplataforma, desenvolvido para a web, que viabiliza o cruzamento de dados obtido por um conjunto de softwares de gerência instalados. Esta nova aplicação, batizada de S.I.M (Sistema Integrado de Monitoramento) tem por objetivo a emissão de alertas, a fim de automatizar e otimizar a resolução de incidentes identificados em uma rede de computadores, auxiliando em sua administração.*

Palavras-chave: *Gerência de rede, monitoramento, segurança da informação.*

1. Introdução

A segurança da informação e de sistemas conectados tem recebido investimentos, superando até mesmo o setor de infra-estrutura física. Esta necessidade de segurança visa preservar e manter os dados seguros contra possíveis ataques ou invasões, identificando possíveis pontos de vulnerabilidades, a fim de implantar medidas preventivas de salva-guarda das informações em um ambiente de rede. Quando uma anomalia é detectada através do NOC (*Network Operation Center*), faz-se necessária a verificação da causa deste incidente. Considerando a existência de diversas ferramentas para gerenciamento de rede e que ao monitorá-la é necessário consultar e analisar relatórios gerados por *softwares* distintos, estas interpretações acabam tornando-se trabalhosas e, em alguns casos, podem ser utilizadas como artifício para desviar a atenção do gerente durante algum processo [CORREA, 1998].

Uma possível hipótese para auxiliar a tarefa do gerente de rede, seria a criação de uma ferramenta que agrupe as diversas informações geradas pelas demais em um ambiente integrado. Baseado nisso, o objetivo deste trabalho é desenvolver um estudo sobre as principais ferramentas livres utilizadas para o monitoramento e controle de ambientes de redes. A partir deste conhecimento, projetar e desenvolver uma solução híbrida que integre as saídas destas ferramentas permitindo desta maneira, uma interação eficiente destas soluções e disponibilizando para o administrador da rede uma aplicação centralizada para o gerenciamento de todos os elementos conectados.

Este artigo apresenta na seção 2 a fundamentação teórica deste estudo, abordando a área de gerência de redes, infra-estrutura de gerência e as principais ferramentas *open source* utilizadas para este fim. Na seção 3 está descrita a metodologia deste trabalho, apresentando como será desenvolvida a aplicação proposta, explicando seu funcionamento e discutindo alguns resultados parciais deste projeto. Na seção 4, estão relatadas as considerações finais sobre o trabalho e por fim, na seção 5, as referências.

2. Gerência de Rede

O gerenciamento de rede está associado ao controle de atividades e ao monitoramento do uso de recursos. As tarefas básicas da gerência em redes são: obter informações da rede, tratar estas informações possibilitando um diagnóstico e encaminhar as soluções dos problemas. Para isso, funções de gerência devem ser embutidas nos diversos componentes de uma rede, para que possibilitem descobrir, prever e reagir a anomalias [DUARTE, 1996].

Um dos modelos criados pela ISO (*International Organization for Standardization*) para uniformizar a gerência de redes foi o FCAPS (*Fault*, *Configuration*, *Accounting*, *Performance and Security*), que se destaca por servir como base para os demais modelos. Este método classifica as áreas funcionais do gerenciamento de redes em cinco categorias: O gerenciamento de desempenho é responsável por quantificar, medir, informar, analisar e controlar o desempenho de diferentes componentes, em virtude de demandas variáveis de tráfego e falhas ocasionais na rede. O gerenciamento de falhas tem o objetivo de detectar e reagir às condições de falhas transitórias da rede. A terceira área é composta pela gerência de configuração que possibilita ao administrador saber quais dispositivos fazem parte do ambiente administrado e quais são suas configurações de *hardware* e *software*. O gerenciamento de contabilização permite que um gerente especifique, registre e controle o acesso de usuários e dispositivos aos recursos da rede. Por fim, a gerência de segurança controla o acesso aos ativos de acordo com alguma política definida [KUROSE e ROSS, 2006].

Nos itens 2.1 e 2.2 serão abordados os componentes do gerenciamento de redes, bem como características e objetivos de suas principais ferramentas que foram estudadas para desenvolver este projeto.

2.1 A infra-estrutura do gerenciamento de rede

O campo do gerenciamento de rede tem sua terminologia específica para os componentes de uma arquitetura de gerência. Esta arquitetura é comporta por três componentes básicos: uma entidade gerenciadora a qual se caracteriza por uma

aplicação executada em uma estação central de gerência da rede e fornece ao seu responsável informações que permitem a análise e identificação de desvio de comportamentos que podem prejudicar o funcionamento do sistema; os dispositivos gerenciados ativos de rede que integram um conjunto de objetos gerenciáveis constituídos por componentes de *hardware* e *software*. Toda informação disponibilizada pelo dispositivo gerenciado é organizada em uma base de dados denominada MIB (*Management Information Base*), que pode ser acessada e modificada pela entidade gerenciadora; e um protocolo de gerenciamento de rede que é executado entre a entidade gerenciadora e o agente de gerenciamento, permitindo que a entidade gerenciadora investigue o estado dos dispositivos gerenciados e, indiretamente, execute ações sobre eles mediante seus agentes [KUROSE e ROSS, 2006]. Para a implementação da solução proposta neste trabalho adotou-se o protocolo SNMP (*Simple Network Management Protocol*), sendo este instalado e ativo em cada dispositivo gerenciado.

Para um monitoramento eficaz dos elementos conectados é necessário o controle do fluxo de informações, o qual pode indiciar a execução de aplicações maliciosas que afetam o comportamento da infra-estrutura da rede, como *worms*, vírus ou até mesmo utilização de programas P2P (*Peer-to-peer*). Desta forma, tem-se no monitoramento de tráfego uma prática essencial para a identificação de comportamentos anômalos [KAMIENSKI *et al.* 2005].

Além do monitoramento, é necessária a utilização de uma solução que capture e analise de forma constante os pacotes e informações que trafegam pela rede, a fim de identificar possíveis vulnerabilidades e tentativas não autorizadas de acesso, assim como atividades ilegais. Um IDS (*Intrusion Detection System*) reúne essas características, tendo a capacidade de detectar atividades suspeitas, impróprias ou de ataques realizados à portas legítimas que não podem ser protegidas por um *firewall* [NAKAMURA e GEUS, 2007]. Por fim, a utilização de uma ferramenta que forneça dados dos *hosts* como aplicações instaladas, em execução, *uptime,* entre outras, completam o conjunto de características necessárias para definir as principais ferramentas de gerência de rede analisadas neste trabalho.

2.2 Principais Ferramentas

As principais ferramentas utilizadas para gerência e monitoramento de ambientes de redes estudadas neste projeto têm como características comuns estarem sob licença GPL (*General Licence Public*) e serem utilizáveis em sistemas baseados em Linux.

O MRTG (*Multi Router Traffic Grapher*), uma ferramenta para coleta de dados que gera gráficos referentes ao tráfego de rede, possibilita o monitoramento do desempenho dos elementos conectados por meio do fluxo de dados de entrada e saída nas portas dos comutadores da rede. Pode ser utilizado na gerência de desempenho, verificando o tráfego dos *hosts* monitorados e na gerência de contabilização onde é verificado o tempo de paralisação dos *host*s na rede [CORREIA, 2004]. Outra ferramenta adotada para este trabalho a qual tem a finalidade de monitorar o desempenho da rede é o Ntop (*Network Traffic Probe*). Algumas de suas funcionalidades são: listar e ordenar o fluxo de dados de acordo com vários protocolos, manter estatísticas permanentemente em banco de dados, identificar passivamente informações sobre os *host*s da rede e decodificar protocolos da camada de aplicação, inclusive os encontrados nos *softwares* tipo P2P [MANN, 2009].

Para auxiliar na gerência de segurança também são utilizadas ferramentas do tipo IDS (*Intrusion Detection System*), que trabalham como um alarme contra as intrusões. Desta forma é possível realizar a detecção baseada em algum tipo de conhecimento, como assinaturas de ataques, ou em desvios de comportamento. Este tipo de aplicativo é capaz de detectar e alertar os administradores quanto a possíveis ataques ou comportamentos anormais no ambiente de rede [NAKAMURA e GEUS, 2007]. Ainda em consonância com os autores citados o Snort é um sistema IDS baseado em regras, capaz de realizar análise de tráfego em tempo real e registro de pacotes em redes IP. Esta aplicação funciona de acordo com uma série de funções que, trabalhando de modo integrado, são capazes de detectar, analisar e responder à atividades suspeitas, podendo ser aplicados para prevenir de forma eficiente incidentes de segurança.

Outra ferramenta no segmento de aplicações para monitoramento é o Nagios, um aplicativo que monitora *hosts* e serviços de rede com o intuito de alertar os administradores sobre possíveis problemas. Este *software* é capaz de identificar serviços e recursos de forma remota. Além do Nagios, outra ferramenta que possui características semelhantes com as que ele apresenta é o Cacti, que por sua vez utiliza scripts em Bash, Perl, XML para coletar informações locais ou remotas através do SNMP. Seus principais scripts medem a latência entre os *hosts*, uso de interfaces de rede, uso do CPU, memória, disco, usuários conectados, etc. [NETO e UCHÔA, 2006].

Com base no estudo destas ferramentas será feita a proposta para o desenvolvimento do S.I.M (Sistema Integrado de Monitoramento), abrangendo algumas das características que cada uma dispõe, reunindo e comparando suas informações em um único ambiente integrado.

3. Proposta de Integração

Para o desenvolvimento deste projeto, as ferramentas escolhidas atendem aos requisitos mínimos de: consentir as exigências da ISO (*International Organization for Standardization*), respeitar as áreas do gerenciamento de redes (gerenciamento de desempenho, de configuração, de contabilização, de segurança e de falhas) e principalmente terem a capacidade de integrarem as informações geradas entre si.

Tabela 1. Comparativo entre Ferramentas

Software	Tipo	Principais características
MRTG	Monitor de Desempenho	Monitora os *hosts* e gera *logs* do tráfego de rede.
Ntop	Monitor de Desempenho	Monitora o tráfego de acordo com diversos protocolos.
Snort	Detecção de Intrusão	Detecta e responde a atividades suspeitas na rede.
Nagios	Monitor de Rede	Disponibiliza informações sobre *hosts* e serviços da rede.
Cacti	Monitor de Rede	Disponibiliza informações sobre *hosts* e serviços da rede.

A Tabela 1 apresenta as ferramentas estudadas e suas principais características, a partir disto foram escolhidas três ferramentas para comporem o novo aplicativo proposto: Ntop, Snort e um *script* em Perl. Este último foi escolhido pelo fato de ser mais simples e objetivo na tarefa de obter informações sobre os *hosts*, dispensando a instalação de outra ferramenta para a obtenção dos mesmos dados. Para realizar a integração das ferramentas definidas, todas elas serão instaladas em um ambiente baseado em Linux e gerarão relatórios sobre o tráfego de rede (Ntop), informações referentes à captura e análise de pacotes (Snort), além de informações sobre os *hosts* (*script* Perl).

3.1 Desenvolvimento da aplicação

A tecnologia adotada para a implementação do *software* é PHP (*PHP Hypertext Processor*), pois esta linguagem de programação permite de forma simples e eficiente a integração dentre o Ntop e Snort. Outra grande vantagem do PHP é a facilidade de visualização e interpretação das informações geradas pelas aplicações de monitoramento, além de possibilitar a disponibilização das mesmas na WEB.

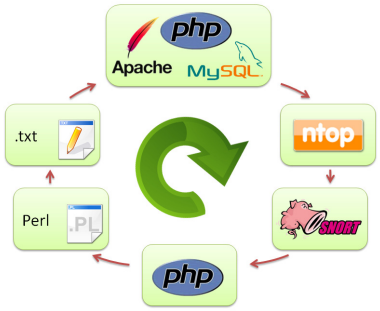

Figura 1. S.I.M – Funcionamento Básico da Aplicação

Para compreender a forma como está proposta a integração apresentada na Figura 1, primeiramente deve-se entender, em baixo nível, como e que informações cada uma delas fornece sobre o ambiente de rede e principalmente a maneira como cada uma delas armazena os dados de seus relatórios.

O Ntop apresenta algumas características pontuais, como o fato de que após instalado, suas configurações só podem ser realizadas através da interface web. Ao ser iniciado, por padrão, roda como *daemon*, disponibilizando um método de consulta as informações sobre os *hosts* na rede, por meio de uma funcionalidade chamada Data Dump, através da qual é possível requisitar dados via comando "wget" do PHP. Através deste comando, o Ntop retorna um *array* com dados sobre os *hosts* presentes na rede. Este *array* possui informações como: nome do *host* na rede, pacotes enviados e recebidos organizados pelos protocolos configurados na execução da ferramenta, entre outras. Essas informações serão utilizados para identificar comportamentos anômalos no ambiente de rede. Um exemplo de retorno do Ntop está apresentado na Figura 2. Assim o Ntop retorna via *Data Dump* um *array* com dados sobre os *hosts* presentes na rede.

A próxima ferramenta a ser utilizada nessa proposta de integração será o Snort, ferramenta IDS utilizada no gerenciamento de rede. Sua instalação requer uma base de dados MySQL, a libpcap que é uma biblioteca utilizada para captura de pacotes na rede e as regras que serão utilizadas no monitoramento.

Assim que instalado, são inseridas e habilitadas às regras da aplicação que servirão para detectar possíveis ameaças a rede, além disso, é configurada a forma de "saída", local onde são armazenadas as informações referentes às ameaças detectadas pela ferramenta. Neste projeto será utilizada a saída via banco de dados MySQL, para isso também será aplicado ao Snort o *plugin* BASE (*Basic Analysis and Security Engine*), pois facilitará a integração com as demais ferramentas devido a melhor organização dos dados armazenados e a facilidade de conexão do PHP, linguagem de desenvolvimento da nova aplicação, com o MySQL.

```
//comando php para requisitar os dados
$foo = system("wget -O ntop.php 'http://localhost:3000/dumpData.html?language=php

//arquivo php com o array de retorno
$ntopHash = array(
'192.168.1.35' => array(
    'hostResolvedName' => 'taciano-pc',
    'pktSent' => '33280',
    'pktRcvd' => '30991',
    'bytesMulticastSent' => '11510',
    'pktMulticastSent' => '87',
    'bytesMulticastRcvd' => '0',
    'pktMulticastRcvd' => '0',
    'bytesSent' => '6546975',
    'bytesRcvd' => '21596361',
    'ipBytesSent' => '6541739',
    'ipBytesRcvd' => '21595809',
    'ipv6Sent' => '0',
    'ipv6Rcvd' => '0',
    'tcpBytesSent' => '6393249',
    'tcpBytesRcvd' => '21372191',
    'udpBytesSent' => '145929',
    'udpBytesRcvd' => '223618',
    'icmpSent' => '2333',
    'icmpRcvd' => '0',
)
);
```

Figura 2. Dados retornados via Ntop

	Campo	Tipo	Collation	Atributos	Nulo	Padrão	Extra
	sid	int(10)		UNSIGNED	Não	None	
	cid	int(10)		UNSIGNED	Não	None	
	signature	int(10)		UNSIGNED	Não	None	
	sig_name	varchar(255)	latin1_swedish_ci		Sim	NULL	
	sig_class_id	int(10)		UNSIGNED	Sim	NULL	
	sig_priority	int(10)		UNSIGNED	Sim	NULL	
	timestamp	datetime			Não	None	
	ip_src	int(10)		UNSIGNED	Sim	NULL	
	ip_dst	int(10)		UNSIGNED	Sim	NULL	

Figura 3. Estrutura de uma tabela do Snort

A Figura 3 representa o local onde ficam armazenadas as ocorrências geradas pelas regras ativas no Snort. A partir dela podem ser obtidas informações como endereço IP de quem fez a requisição, bem como o de destino, a data e hora (*timestamp*) em que aconteceu o evento, uma descrição do evento, entre outras. Estes dados serão utilizados na integração proposta pela nova aplicação.

Por sua vez, o *script* em Perl (Figura 4) extrai as informações de dados do dispositivo gerenciado através do serviço SNMP. Este serviço é responsável pelo gerenciamento e respostas das requisições de informações do protocolo de gerenciamento no servidor, que busca os dados na MIB. Para este tipo de requisição o serviço de SNMP deve estar instalado e ativo no dispositivo gerenciado de destino.

```
@PhysAddress = snmpwalk ("-v 1 -c public $_ ifPhysAddress ");
@Name = snmpwalk ("-v 1 -c public $_ sysName ");
@Date = snmpwalk ("-v 1 -c public $_ hrSystemDate ");
@UpTime = snmpwalk ("-v 1 -c public $_ sysUpTime ");
@Installed = snmpwalk ("-v 1 -c public $_ hrSWInstalledName ");
@Run = snmpwalk ("-v 1 -c public $_ hrSWRunName ");
```

Figura 4. Comandos Perl utilizados para obter informações via SNMP

3.2 Funcionamento da aplicação

O ponto de partida para o funcionamento da ferramenta que está sendo proposta será o monitoramento do tráfego de rede realizado pelo Ntop e a captura de pacotes do Snort, conforme a Figura 5.

A *baseline* é gerada via "*Data Dump"* do Ntop que a define de acordo com a média de tráfego do *host*. Sempre que for percebida alguma anomalia no tráfego do ativo monitorado, o S.I.M buscará informações mais detalhadas sobre o que está acontecendo através do relacionamento das informações de consumo geradas pelo Ntop com a captura de pacotes do Snort. Em seguida, é enviado o IP do *host* envolvido na anomalia como parâmetro para a chamada de execução do script Perl, que por sua vez buscará as informações como nome da máquina, MAC da interface de rede, *uptime* e processos em execução, gerando um arquivo .txt que é interpretado pelo S.I.M. Por fim, todas as informações referentes ao incidente detectado são armazenadas na base de dados da ferramenta.

O S.I.M deve fornecer ao administrador da rede e também a equipe de suporte aos usuários um conjunto de informações relevantes, a fim de auxiliar na solução de problemas ou anomalias identificadas no ambiente de rede da organização. A resposta rápida a esses incidentes está diretamente relacionada ao tempo despendido na detecção dos mesmos, sendo desta forma imprescindível para a equipe de TI receber informações rapidamente, diminuindo assim o tempo de resposta no tratamento das causas identificadas.

A aplicação apresenta uma solução simples para a questão da comunicação dos incidentes detectados: primeiramente o S.I.M se encarregará de criar uma base de das anomalias identificadas e juntamente com o armazenamento dos dados, e-mails de alertas serão disparados para o administrador da rede e responsáveis pela área de TI. Por fim, o administrador da rede acessa o sistema e visualiza todas as informações referentes aos incidentes detectados, podendo informar a medida adotada em cada caso, a partir dessas informações, possa ser criada uma base de conhecimentos sobre as anomalias da rede para ser utilizada futuramente.

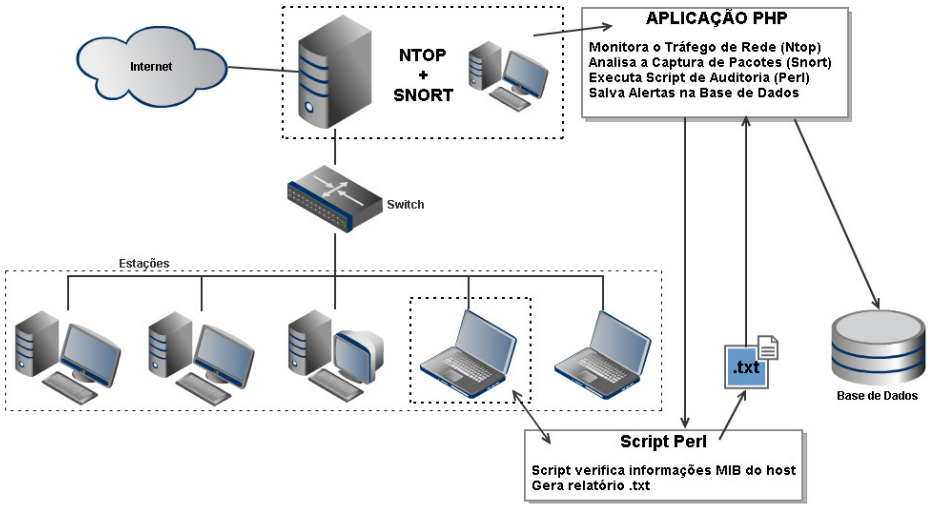

Figura 5. S.I.M – Arquitetura Básica da Aplicação

3.3 Resultados Parciais

Este projeto já está em fase de desenvolvimento, até aqui sua estrutura conta com a interação entre o Ntop, Snort e *script* Perl funcional, operando de forma automática via *crontab*. As informações geradas pelo S.I.M poderão auxiliar o gerente na busca por anomalias na rede, visto que, quando o tráfego de rede foge ao padrão que o *host* em questão possui, a ferramenta pesquisa no Snort os alertas que aquele ativo gerou no mesmo espaço de tempo em que o tráfego foi considerado anormal. As informações integradas são apresentadas ao administrador de rede, com isso, ele possui maior conhecimento sobre o que e onde estão ocorrendo possíveis problemas.

Com a visualização integrada proporcionada pelo S.I.M. tem-se um grande impacto sobre a gerência da rede, pois com estas informações o gerente pode tomar decisões mais concisas, tendo em vista que esta ferramenta facilita a análise do possível incidente de rede. Outra vantagem da análise através da ferramenta central é a redução do número de possíveis falsos positivos gerados pela análise de um *software* isolado, problemas de sincronismo de horários entre os servidores ou mesmo erro de interpretação do gerente ao utilizar multi telas para comparação.

Na Figura 6 é apresentada a tela principal do S.I.M, mostrando as anomalias identificadas, o range de IP's que a ferramenta está configurada para trabalhar e também uma lista com os *hosts* ativos na rede. O sistema apresenta as anomalias dividindo-as em cores, vermelho para as que ainda não foram averiguadas, amarelo para as que estão em análise e verde para as que já foram analisadas pelo administrador da rede.

Figura 6. S.I.M – Print screen da ferramenta

4. Considerações Finais

Os objetivos deste projeto estão sendo gradualmente alcançados, através dos resultados obtidos com a implementação e testes do S.I.M. Tais resultados demonstram que é possível agilizar o processo de detecção de anomalias em uma rede de computadores, através da sinergia e do cruzamento de informações relevantes sobre o funcionamento da rede, e da disponibilização destas aos administradores do ambiente. O S.I.M proporciona, com suas atividades de monitoramento do fluxo da rede, captura de pacotes e da intersecção destas informações, uma redução no tempo de detecção e tratamento das anomalias em um ambiente conectado e distribuído. Além disso, a base de conhecimento, implementada a partir do registro de dados sobre as anomalias, disponibiliza uma amostragem que poderá ser utilizada em trabalhos futuros.

5. Referências

CORREA, Claudio (1998). "Detecção de Ataques em Redes de Computadores", http://www.das.ufsc.br/gia/pb-p/rel-claudio-00/relatorio.html, Maio.

CORREIA, Marcelo (2004), "Gerência de Redes", http://www.ccet.unimontes.br/arquivos/dcc/gilmara/1144.pdf, Maio.

DUARTE, Otto M.B. (1996). "Gerenciamento de Redes", http://www.gta.ufrj.br/~alexszt/ger/snmpcmip.html#sec1a, Maio.

KAMIENSKI, Carlos; SOUZA, Tatiane; FERNANDES, Stenio; SILVESTRE, Guthemberg; SADOK, Djamel. "Caracterizando Propriedades Essenciais do Tráfego de Redes Através de Técnicas de Amostragem Estatística", SBRC 2005.

KUROSE, James F.; ROSS, Keith W. (2006). "Redes de Computadores e a Internet Uma Abordagem Top-Down", São Paulo, Edição 3.

MANN, Cesar (2009). "Análise Constante", http://ppgia.pucpr.br/~jamhour/Download/pub/RSS/MTC/referencias/TCC-2009.pdf, Junho.

NAKAMURA, Emilio T.; GEUS, Paulo L. (2007). "Segurança de Redes em Ambientes Cooperativos", São Paulo, Edição 1.

NETO, Arlindo; UCHÔA, Joaquim (2006). "Ferramentas Livres Para Monitoração de Servidores", http://www.ginux.ufla.br/files/artigo-ArlindoNeto,JoaquimUchoa.pdf, Junho.