Mauricio Santos Teixeira

Network Discovery
Técnicas e Ferramentas

Monografia de Pós-Graduação "*Lato Sensu*" apresentada ao Departamento de Ciência da Computação para obtenção do título de Especialista em "Administração em Redes Linux"

Orientador
Prof. Joaquim Quinteiro Uchôa

Lavras
Minas Gerais - Brasil
2004

Mauricio Santos Teixeira

Network Discovery
Técnicas e Ferramentas

Monografia de Pós-Graduação "*Lato Sensu*" apresentada ao Departamento de Ciência da Computação para obtenção do título de Especialista em "Administração em Redes Linux"

Aprovada em *12 de Dezembro de 2004*

Prof. Gustavo Guimarães Parma

Prof. Ricardo Martins de Abreu Silva

Prof. Joaquim Quinteiro Uchôa
(Orientador)

Lavras
Minas Gerais - Brasil

Agradecimentos

Ao Prof. Joaquim Quinteiro Uchôa por ter sido um guia tão importante no decorrer do curso e mais ainda nesse momento decisivo.

Aos demais professores do curso de Administração de Redes Linux, que tanto contribuíram com seus conhecimentos sem nunca pedir nada em troca. Em especial aos Profs. Bruno de Oliveira Schneider e Ricardo Martins de Abreu Silva, com quem muito me identifiquei e aprendi.

À minha esposa, Virgínia, que muito me cobrou, ouviu e ajudou. Ter uma esposa que também é da área pode ser muito estimulante.

Resumo

O gerenciamento de redes compreende vários processos em diversas áreas da informática. Um deles trata de documentar e gerenciar recursos e máquinas. O *network discovery* é uma parte desse processo, e consiste na localização de elementos e/ou serviços que compõem uma rede. Este trabalho direciona o leitor no sentido de conhecer as principais técnicas e algumas das ferramentas disponíveis para o auxílio nesse processo. Alguns tópicos já são de conhecimento de boa parte dos administradores de rede, mas recebem um direcionamento no contexto da coleta de informações com o fim benigno de criar uma consciência gerencial.

Aos meus pais, que sempre acreditaram em mim e investiram seu tempo, dinheiro e muitas noites mal dormidas para eu ser quem eu sou.

“O que não se mede, não se administra; o que não é mensurável, faça-o mensurável”
Galileu Galilei, físico e astrônomo italiano.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Esta monografia tem por objetivo demonstrar algumas das diversas técnicas e ferramentas existentes para a localização de elementos ativos em uma rede de computadores. O conjunto dessas técnicas é conhecido como "*network discovery*", por tratar-se de procedimentos utilizados para "descobrir o que existe" em uma rede. A intenção não é demonstrar todas as técnicas, nem criticá-las ou exaltá-las, mas apenas demonstrar algumas delas, bem como algumas aplicabilidades, vantagens, desvantagens e ferramentas que se utilizam das mesmas. Apesar do claro uso desse conteúdo como facilitador de atividades ilícitas (como invasão de sistemas), o objetivo específico deste trabalho é apresentar o *network discovery* como uma parte importante na vida de um administrador de redes ou de um consultor de informática, no sentido de ajudá-lo a levantar mais informações e por conseguinte propiciar um melhor gerenciamento da rede.

Pensando nas questões que motivam este trabalho, existem muitos pontos de vista que podem ser utilizados para justificá-lo, bem como todas as outras questões que o envolvem. Dos principais fatores que contribuem para a necessidade de conhecer as técnicas e ferramentas aqui citados destacam-se dois: terceirização e gerenciamento.

No contexto da terceirização percebemos as modalidades interna (mais comum, onde o serviço é executado dentro da empresa) e externa (também conhecida como *outsourcing*, onde o serviço é executado na estrutura do parceiro). Na interna o serviço é contratado por uma empresa externa, mas esta aloca um ou mais funcionários e/ou recursos no ambiente do contratante, e todo o serviço é executado localmente (esse é o modelo mais comum). Já com relação à terceirização externa (também conhecida como *outsourcing*) todo o serviço é executado em um ambiente externo ao contratante, normalmente dentro da infra-estrutura do contratado, que só vai ao cliente em reuniões regulares para discutir resultados.

Atualmente, muitas empresas têm optado pela terceirização externa, cada uma pelos motivos mais diversos. Algumas tomam sua decisão com base nos custos de manutenção mensal (menor em alguns casos), outras nos custos de longo prazo (como treinamento, manutenção de pessoal, atualização tecnológica, etc).

Este trabalho não pretende justificar ou direcionar a escolha por nenhuma modalidade. A questão é que a terceirização implica em documentar e conhecer as redes com a maior precisão posssivel, seja por parte do técnico que irá implantar, do gerente que irá contratar, ou até mesmo de outro profissional que será responsável pela manutenção do produto (ou serviço) final.

Supondo seguinte panorama: um técnico é contratado para fazer uma análise de uma rede, onde pouca coisa está documentada. Essa rede é muito grande, e abrange localidades geograficamente distantes (uma WAN). O consultor precisa conhecer a rede como um todo antes de poder expressar qualquer opinião sobre a mesma. Dessa forma, o conhecimento doravante apresentado neste trabalho é justificado sob a ótica de que ele será utilizado para localizar todos os equipamentos e elementos dessa rede (passivos e/ou ativos), bem como sua topologia, distribuição e capacidade atual.

Tal exemplo não se aplica apenas a grandes empresas, pois hoje em dia é possível encontrar empresas muito menores com estruturas tão complexas quanto a apresentada. Suponha-se uma locadora de vídeo que possui duas filiais em bairros diferentes da cidade, onde as mesmas comunicam-se através de VPNs (*Virtual Private Networks*) utilizando acesso ADSL (*Asymmetric Digital Subscriber Line*). Ou o caso de um escritório de profissionais liberais, onde o consultor é chamado com urgência, mas ninguém com conhecimento suficiente da rede está disponível (ou não existe).

Todo o raciocínio apresentado também tem um outro lado, o do gerente, que contrata uma empresa e/ou um consultor para implantar uma nova rede, alterar a existente, ou implantar um serviço externo ao seu ambiente (*outsourcing*). Após concluído o serviço, o gerente precisa validar o produto final, e portanto precisa de uma análise que transceda a documentação entregue (apesar de que a entrega de documentação é uma prática pouco comum, e por vezes muito imprecisa).

Tal visão do uso das técnicas como meio de validação também serve como o início do processo seguinte, que é o gerenciamento. Muitas vezes, no entanto, as técnicas aqui apresentadas serão aplicadas em redes já existentes no sentido de melhorar ou até mesmo de criar uma documentação que talvez nem exista. O objetivo desse processo, então, seria o de confirmar se a rede realmente possui tantos e quais elementos pensa-se ter, de forma a garantir um monitoramento e manutenção adequados.

É muito comum, hoje em dia, empresas ou profissionais que implantam solu-

ções de rede das mais simples às mais complexas, mas vão postergando o processo de documentação até um ponto em que foge do controle. Chega-se o momento em que a rede está toda implantada, totalmente operacional, mas perdeu-se o controle sobre quantos pontos (estações) estão ativos, quais circuitos de acesso estão roteando para quais redes, etc. Às vezes essa documentação não é feita porque a própria pessoa que implanta é aquela que vai gerenciar, e essa não sente a necessidade de documentação. Contudo, em algum momento, é possível que essa pessoa seja substituída, seja por uma fatalidade ou necessidade, e o próximo a assumir não tenha a oportunidade de receber nenhuma informação sobre a estrutura, nem mesmo através de conversas informais.

Ainda relacionado à gerencia de rede, as técnicas deste trabalho podem ser utilizadas para verificação de um dos grandes problemas atuais: segurança. É possível utilizá-las para verificação da facilidade de localizar os elementos de uma rede, seja internamente ou externamente, o que pode facilitar o trabalho de um possível invasor. Também, muitas das técnicas são contempladas pelos sistemas de detecção de intrusão (IDS, *Intrusion Detection Systems*), e portanto podem ser utilizadas para validar o seu funcionamento.

Esses são apenas alguns dos argumentos que podem justificar este trabalho. Cada empresa ou pessoa pode encontrar o seu. O importante é ter sempre em mente que um dos pontos chave para a boa administração de qualquer rede é conhecê-la o mais detalhadamente possível.

Capítulo 2

Conceitos Básicos

Ao longo deste trabalho será utilizado o termo *"elemento"* para identificar qualquer componente de uma rede. Apesar de não haver encontrado referências bibliográficas que reforcem o uso do termo, o mesmo foi escolhido pois parece ser mais adaptável tanto à idéia de *hosts* (estações e/ou servidores) e qualquer outro componente passivo ou ativo em uma rede (roteadores, *gateways*, *switches*, etc). Os termos específicos (*hosts*, *firewalls*, etc) também serão utilizados quando necessário.

Outro termo importante que precisa ser previamente esclarecido é *"RFC"* (*Request For Comments*, RFC 2026 [BRADNER]). Trata-se de documentos cujo objetivo é explicar, demonstrar e/ou documentar procedimentos e/ou protocolos de redes (normalmente da Internet). Cada RFC publicada recebe um número e fica disponível publicamente por um período onde qualquer pessoa pode questioná-la, criticá-la e/ou implementá-la e provar o conceito difundido. Durante esse tempo, uma RFC pode acabar tornando-se um padrão de mercado por ter sido amplamente difundida, tornado-se útil e bastante aceita. Sendo assim, após um período (não determinado), essas RFCs podem tornar-se STD (*Internet Official Protocol Standard*), e portanto desse ponto em diante precisam ser seguidas à risca. Outras RFCs não propõem nada novo, apenas a melhor maneira de implementar princípios ou conclusões sugeridas como opcionais em outras RFCs ou STDs, e passam a ser conhecidas como BCP (*Best Common Practice*). Durante este trabalho, todos os documentos consultados serão referenciados como RFC (como são publicados), mesmo sendo STD ou BCP.

2.1 Network Discovery

Network discovery (*descobrimento de rede*, ou *localização de elementos em uma rede*) pode ser entendido como o conjunto de técnicas, aplicadas através de ferramentas de *software*, utilizadas no sentido de localizar e documentar os diversos componentes de uma rede de computadores (local ou geograficamente distribuída).

Os protocolos de rede são divididos em diversas camadas de comunicação. O descobrimento de elementos em uma rede pode ser aplicado a qualquer uma dessas camadas, isoladamente ou em conjunto. Existem diferentes dispositivos e protocolos que atuam em uma ou mais camadas. Sendo assim, no momento em que for escolhida a ferramenta ou técnica, é necessário determinar qual o nível e tipo de informação desejado. Seja qual for a camada analisada (ou mais de uma), a intenção é localizar todos os elementos que podem gerar tráfego, erros ou congestionamento na rede, sejam eles elementos ativos ou passivos.

No que diz respeito à implementação em camadas dos protocolos, elas podem variar em quantidade, objetivo e recursos, de acordo com o modelo seguido. Neste trabalho o foco será direcionado às camadas 2 (rede, protocolo IP) e 3 (transporte, protocolos TCP e UDP) da implementação conhecida como TCP/IP (*Transmission Control Protocol*, *Internet Protocol*), já que o mesmo tem se destacado como o padrão de mercado, principalmente por causa da Internet. Também serão análisados alguns protocolos que fazem a ligação entre as camadas, como o ARP (enlace e rede, nas camadas 1 e 2) e o DNS (transporte e aplicação, nas camadas 3 e 4).

No decorrer do trabalho também serão permitidos comentários ou até mesmo análises de protocolos em outras camadas, principalmente na camada 1 (enlace), como o *Ethernet*. Além disso também serão feitas referências ao BGP, OSPF e RIP (camada 2, rede), bem como o SNMP (camada 4, aplicação).

2.1.1 Domain Topology Discovery

O descobrimento da topologia de um domínio da rede trata da localização de dispositivos em um ambiente restrito, como uma LAN (não importando a dimensão). Tal análise pode ser considerada a mais simples, ou mais complexa dependendo, do nível de profundidade requerido.

É possível localizar apenas estações, servidores e serviços (mais simples), ou aprofundar o processo localizando equipamentos de interconexão (*hubs*, *switches*, roteadors, *access points* e *bridges*).

Os domínios de rede podem subdividir-se em *físicos* e *lógicos*. Domínios físicos são todas as máquinas que compartilham o mesmo barramento de rede, ou seja, a mesma localização física, *switch*, etc. Já os domínios lógicos são máquinas que compatilham de um mesmo esquema de endereçamento e/ou localização

(sub-redes). Vários domínios lógicos podem estar contidos em um único domínio físico (mesma estrutura de cabeamento), assim como um único domínio lógico pode estar distribuído entre vários domínios físicos (separados por circuitos de comunicação, pontes, etc).

A comunicação entre os diversos domínios lógicos se dá através dos *gateways* (roteadores e *firewalls*), e os diversos domínios físicos conectam-se através de concentradores como *hubs*, *switches* ou até mesmo *bridges*.

2.1.2 Backbone Topology Discovery

O descobrimento da topologia de *backbone* foca na localização de elementos que compõem uma rede WAN, pois intenciona a identificação principalmente de roteadores e os diversos *links* de comunicação associados aos mesmos.

Esse tipo de identificação trabalha apenas nas camadas 3 e 4, dado que o acesso ao nível físico é restrito, ou inexistente. Aqui são analisados principalmente os *gateways* (roteadores e *firewalls*).

2.2 Network Scan

O processo de varredura de uma rede consiste em tentar localizar elementos através de "tentativa e erro". Não que esse processo seja uma variação do *network discovery*, mas sim uma parte do mesmo, amplamente utilizado por ser simples e ter muitos *softwares* disponíveis.

Diariamente muitos administradores de rede utilizam essas ferramentas com o objetivo de verificar quais estações estão ativas, se seus *links* estão operacionais, se alguns serviços estão disponíveis na rede, etc.

As técnicas utilizadas nesse processo também serão citadas ao longo deste trabalho, já que fazem parte do processo maior de descobrimento da rede, e podem ser utilizadas como uma forma de simplificar certas tarefas, ou até mesmo de conseguir o primeiro "rascunho" da rede a ser analisada.

Capítulo 3

Análise de Técnicas

Nesse capítulo serão analisadas as técnicas de *network discovery*, que são o objetivo principal deste trabalho. Conforme apresentado anteriormente, existem dois grandes grupos de técnicas. No âmbito do *Domain Topology Discovery*, iremos analisar as técnicas relacionadas a PING (sequencial e *broadcast*), *TCP/UDP Scan*, *ARP* (*scan* e *table reading*) e o *DNS Table Query*. Já com relação ao *Backbone Topology Discovery*, serão analisados o *Routing Table Reading*, *Traceroute* e *Dynamic Routing*. Também será feita uma referência ao *Passive Discovery*, que é uma técnica mista.

3.1 PING

Algumas das técnicas envolvem o uso de PING. Antes de demonstrá-las, precisamos esclarecer seu conceito.

De acordo com a RFC 2151 [KESSLER/SHEPARD], PING é um acrônimo para *Packet Internetwork Groper* (uma tradução aproximada, seria o *"Apalpador" de Pacotes Inter-rede*), mas o autor do comando diz que a idéia veio do som emitido por um sonar. Considerado um recurso básico para determinar a disponibilidade de um *host*, ele consiste em dois tipos específicos de pacotes ICMP (*Internet Control Message Protocol*, RFC 792), que são pacotes simples e pequenos, transmitidos diretamente agregados à camada IP, conforme podemos verificar abaixo:

Tabela 3.1: Datagrama IP com ICMP

End. Destino	End. Origem	Tipo	Cabeçalho IP	**Cabeçalho ICMP**	Dados	CRC

Tabela 3.2: Cabeçalho ICMP

Tipo	Código	*Checksum*
Identificação		Sequência

O funcionamento básico do PING consiste no envio de uma mensagem de solicitação (*ICMP ECHO REQUEST*, ou *PING*, pacote com código 8), com três informações principais: tempo de vida (TTL, *Time To Live*, normalmente 255, cabeçalho IP), identificação do pacote (cabeçalho ICMP) e *host* de destino (datagrama). Cada *gateway* que recebe o pacote decrementa em uma unidade o TTL. Se o valor for maior que zero, roteia para o *host* seguinte (outro *gateway*, ou o *host* de destino). Se o TTL for igual a zero, o *gateway* retorna uma mensagem de *Time Exceeded* (tempo de vida expirado em trânsito). Se não for possível rotear, é retornada a mensagem de *Destination Unreachable* (destino inalcançável). Quando o pacote chega ao destino, esse retorna um pacote de resposta da solicitação (*ICMP ECHO REPLY*, ou *PONG*, pacote com código 0), contendo as mesmas informações (um novo TTL, a identificação do pacote de solicitação, e o endereço de destino que, no caso, é o endereço do solicitante). Esse mesmo pacote passa por todo o processo novamente.

Apesar de ser um processo simples, e aparentemente à prova de falhas, alguns problemas são esperados.

Primeiramente, no momento em que o pacote de solicitação é enviado na rede, o *host* de origem inicia um temporizador progressivo, nesse exemplo com o tempo máximo de 2000 milisegundos. Se nenhuma resposta for recebida nesse período, o *host* é considerado inativo. Tal temporizador evita que o *host* de origem fique aguardando indefinidamente por um pacote, mas ao mesmo tempo cria um ponto de falha pois o tempo entre o transporte do pacote de solicitação ser enviado e o pacote de resposta ser recebido não poderá ultrapassar o tempo especificado. Vencido esse prazo, mesmo que um pacote de resposta seja recebido ele não mais será interpretado, pois o *host* de origem já o terá ignorado, classificando como perdido. Um dos itens que pode causar esse tempo de resposta elevado é o fato que os pacotes de PING são tratados como dados de baixa prioridade nos roteadores, e portanto podem ter sua transmissão postergada até o limite de torná-la inviável (do ponto de vista do *host* de origem), ou até mesmo descartados da fila de transmissão.

De acordo com a RFC 1122 (*Requirements for Internet Hosts - Communication Layers* [BRADEN]) cada *host* é **obrigado** a implementar a função de *Echo server*, ou seja, uma função que pode solicitar e responder aos PINGs. Contudo, essa mesma RFC também define que os *hosts* **deveriam** também implementar um

recurso de aplicação para que seja possível ativamente enviar e responder às requisições para fins de diagnóstico. Contudo, existe uma contradição expressa no item 3.2.2.6 da RFC 1122 [IETF]

> *"essa posição neutra resulta em um expressivo debate entre aqueles que acham que o ICMP Echo para endereços de broadcast proporcionam uma valorável capacidade de diagnóstico, e aqueles que acham que o mal uso desse recurso pode facilmente gerar uma tempestade de pacotes"* (tradução livre do item 3.2.2.6 da RFC 1122).

Não existe nenhum documento oficial (RFC) que reforce a idéia do bloqueio de pacotes ICMP, ou seja, qualquer um que o faça estará indo contra as normas. Contudo existem diversos documentos (formais ou informais) que levantam questões sobre como o ICMP é inseguro, e pode ser utilizado para invasões, localizações indevidas, ataques de negação de serviço, e muitos outros problemas de segurança (vide [FARROW] e [HIPPY]). Contudo, essa não conformidade gera outros inconvenientes ao processo de *network discovery*, pois não temos como saber se um determinado *host* não está respondendo porque não pode (indisponibilidade) ou porque não quer (opção por não responder).

Outro problema conhecido é o fato de que muitas empresas bloqueiam o encaminhamento desse serviço na sua rede através de filtros nos *firewalls* e/ou roteadores, a fim de evitar o *PING flood* e o *PING of death*. O *PING flood* é um tipo de DoS (*Denial of Service*, ou Negação de Serviço) que consiste na inundação de pacotes direcionados a um *host* ou rede, com o objetivo de sobrecarregar um *link* de acesso ou o processamento de um *host*. Já o *PING of death* (PING da morte, ou *teardrop*) consiste no envio de um pacote de PING muito grande (acima de 65.536 bytes), maior que o permitido em um pacote TCP/IP. Com relação a esse último, apesar de existirem correções bastante divulgadas para todos os sistemas operacionais, e até mesmo limitações na implementação do comando PING (não deixando definir tamanhos longos de pacote), ainda assim é importante esse tipo de bloqueio pois nunca se conhece ao certo a situação do cliente nem o nível de conhecimento do atacante.

3.1.1 PING Sequencial

O PING sequencial em uma rede (também conhecido como *PING scan*, ou *varredura por PING*), consiste em enviar um pacote PING para cada endereço IP que compõe uma subrede especificada, ou uma lista de endereços IP determinados. O nível de precisão dessa técnica está diretamente associada aos problemas demonstrados anteriormente na análise do PING.

Uma das vantagens dessa técnica reside no fato de que a ferramenta já está disponível. O comando `'ping'` já faz parte da distribuição básica de qualquer sistema operacional, e a especificação manual de cada endereço pode ser facilmente substituida por scripts para automatizar essa tarefa. Além dessa ferramenta simples, muitas outras ferramentas também estão disponíveis, através de arquivos pequenos e funcionamento simples, que podem ser gratuitamente copiadas da Internet.

Em contrapartida à facilidade de uso e disponibilidade de ferramentas, esses pacotes podem ser bloqueados (conforme visto anteriormente) ou até mesmo interpretados e detectados (e usados em auditorias). Os *firewalls* e sistemas de IDS (*Intrusion Detection System*) podem ser configurados de forma a detectar ocorrências de PING sequencial. Muitas vezes esse recurso está disponível em uma rede pois o PING sequencial é muito utilizado por usuários maliciosos da Internet que buscam máquinas ativas em uma rede para executar as mais diversas operações danosas (invasão de sistemas, infecção por vírus, sobrecarga, etc).

Outra desvantagem é o uso dessa técnica como ferramenta isolada, pois ela não consegue levar em conta a topologia física da rede como um todo. Isso significa dizer que se o PING é feito em uma única subrede, mesmo que haja uma bridge fazendo *link* com outras redes externas, do ponto de vista da ferramenta tudo haverá uma única. E mesmo que a busca seja feita em várias subredes, o PING em si não sabe determinar quem é roteador, que redes são locais ou remotas, etc, o que causa uma grande imprecisão no resultado quando se quer conhecer a estrutura física da rede. Mesmo que a intenção seja determinar apenas quantas ou quais máquinas existem na rede, nem todas elas respondem ao PING, como discutido anteriormente.

3.1.2 Broadcast PING

Cada subrede TCP/IP possui dois endereços de uso reservado. Usando como exemplo a rede 192.168.0.0/24, os endereços reservados são o 192.168.0.0, que é o *endereço de rede* (indentifica onde a rede inicia) e o 192.168.0.255, que é o *endereço de broadcast* (identifica onde a rede termina).

O termo *"broadcast"* (difusão) é utilizado pois, de acordo com o item 3.3.6 da RFC 1122 [IETF], existem quatro formatos de endereços de *broadcast* (limitado, direcionado, direcionado a subrede e direcionado a todas as subredes) e que "um *host* DEVE reconhecer cada um desses formatos no endereço de destino de um datagrama recebido". Ou seja, o envio de um pacote a um endereço de *broadcast* significa estar enviando a todos os hosts componentes de uma dada subrede. No entanto, o item 3.2.2.6 da mesma RFC define que "uma requisição de ICMP echo

enviada para um endereço de *broadcast* (...) PODE ser silenciosamente descartada".

Esta última afirmação tem sido alvo de muitas discussões ao longo dos anos. Os desenvolvedores dos sistemas operacionais, especialistas de rede e demais interessados, têm amplamente discutido se esse tipo de pacote deve ser ignorado. Muitos deles concordam de maneira positiva e já implementaram em seus sistemas ou equipamentos mecanismos para só aceitar um PING no endereço de *broadcast* quando esse comportamento for expressamente desejado pelo administrador do sistema, ou seja, especificado em um item de configuração.

Dois dos problemas que contribuiram para esse comportamento foram o *broadcast storm* e os ataques do tipo *smurf*. O *broadcast storm* ocorre quando uma mensagem é enviada para o endereço de *broadcast* de uma rede, e a resposta dessa mensagem também é enviada para o endereço de *broadcast*, o que vira uma bola de neve e bloqueia completamente o tráfego (intencionalmente ou não). Já o *smurf* é um outro tipo de DoS, que consiste no envio de um pacote PING para o endereço de *broadcast* de uma rede, porém com o endereço de origem forjado de forma que esse aparente ser o endereço de uma vítima, ou seja, todas as máquinas de uma subrede então respondem o PING para esse endereço, criando um *PING flood*, sobrecarregando o destino ou os meios de comunicação utilizados para alcançar o mesmo.

Face ao que foi exposto nesse item, essa técnica apenas possui um comportamento diferente da técnica descrita anteriormente (PING sequencial), mas sofre dos mesmos problemas, além de alguns problemas próprios.

3.2 TCP/UDP Scan

Alguns elementos da rede não respondem a requisições de PING, conforme vimos nos itens anteriores, mas ainda assim sabemos (ou não) que estão ativos e provavelmente executando algum tipo de serviço em rede. Diante desse ponto de vista, ao invés de procurar pela existência ou não desses elementos, podemos procurar pelos serviços públicos que eles executam (através de uma busca por portas abertas), e havendo resposta de um, entendemos que ali temos um elemento ativo.

Contudo, essa solução sofre dos mesmos problemas citados anteriormente com relação ao bloqueio em *firewalls* e aos sistemas de detecção de intrusos. De uma maneira geral um *firewall* irá bloquear conexões de entrada em qualquer porta que não esteja relacionada a um serviço público. Ou seja, a detecção só será possível em elementos da rede que notadamente possuem algum serviço cujo acesso será liberado, impedindo a detecção de qualquer outro elemento que não o seja.

Seguindo essa linha de raciocínio, podemos imaginar um escritório com acesso

a internet, cujo site e servidor de e-mail sejam completamente terceirizados em *datacenters* externos. Esta é uma situação onde o *firewall* serve apenas para conexão externa, impossibilitando a detecção das estações componentes da rede. Contudo, se por outro lado estivermos dentro da mesma, as estações poderão também possuir *firewalls* pessoais, já que elas são apenas estações de trabalho que não servem nenhum recurso à rede, de maneira que só iremos detectar o servidor. E mesmo assim, esse servidor poderá possuir um sistema de detecção de intrusão reativo (que reage às ocorrências executando alguma ação preventiva ou corretiva), que detectará a busca pelas portas e imediatamente criará um bloqueio, gerando dúvidas quanto à disponibilidade ou não do elemento.

3.3 ARP

Algumas das técnicas envolvem o conceito de ARP. Antes de demonstrá-las, precisamos esclarecer esse conceito.

O ARP (*Address Resolution Protocol*), definido pela RFC 826 [PLUMMER] (e reforçado pela RFC 894 [HORNING]) é um protocolo utilizado pela pilha TCP/IP, que opera na camada 2 (rede), que serve para converter endereços IP em endereços físicos (camada 1) para transmissão de dados em uma rede *Ethernet*. Assim como em uma rede TCP/IP, cada elemento de uma rede *Ethernet* possui um endereço único que o identifica. Esta ligação entre os dois endereços pode ser feita de maneira estática (através de tabelas de conversão cadastradas em cada elemento da rede) ou dinâmica. Contudo o procedimento estático se torna bastante complexo em redes com um grande número de elementos (principalmente se uns não precisam falar com todos), ou naquelas que mudam com certa frequência.

Tabela 3.3: Pacote ARP

Tipo de *Hardware*		Tipo de Protocolo
Tam. do End. de *Hardware*	Tam. do End. do Protocolo	Cód. da Operação
Endereço de *Hardware* de Origem		
Endereço do Protocolo de Origem		
Endereço de *Hardware* de Destino		
Endereço do Protocolo de Destino		
Dados		

A resolução de endereços através do ARP tem início quando o elemento T deseja transmitir uma informação para o destino R, é enviada um uma mensagem de solicitação ARP para o endereço de *broadcast* da rede *Ethernet*, no formato `who is R.R.R.R tell T.T.T.T` (ou seja, “quem possuir o endereço

R.R.R.R, responda ao endereço T.T.T.T"). Como todos os elementos da rede recebem essa informação, existe uma alta probabilidade de que o elemento correto responda à mesma (considerando que ele está no mesmo segmento) com uma mensagem no formato `R.R.R.R is ee:ee:ee:ee:ee:ee` (ou seja, "o endereço IP R.R.R.R é conhecido pelo endereço *Ethernet* ee:ee:ee:ee:ee:ee"). O pacote de resposta é enviado diretamente ao elemento que enviou a requisição. O elemento R imediatamente já aprende o endereço de origem T, pois o pacote de requisição já contem o seu endereço. Ambos conhecendo os endereços envolvidos, a transmissão pode ser iniciada.

Concluído o processo de resolução, ambos os elementos T (origem) e R (destino) armazenam os endereços dos pares envolvidos em um tabela local de mapeamento estático. Esta tabela serve para reduzir a necessidade de novas pesquisas de endereços na rede, já que o envio de muitos *broadcasts* gera um tráfego desnecessário a qualquer operação normal. No entanto essa tabela tem tamanho finito, e portanto precisa ser constantemente revisada de forma a remover endereços que não foram utilizados após um determinado tempo. Além disso, alguns elementos podem não mais existir na rede após um certo período de tempo, e um novo elemento tomar seu lugar com o mesmo endereço IP mas, obviamente, com um endereço físico diferente. Dessa forma, o item 2.3.2.1 da RFC 1122 [IETF], entitulada *"ARP Cache Validation"* sugere vários métodos a serem utilizados (isoladamente ou em conjunto) para a expiração de entradas nessa tabela. A tabela de endereços gerada pelo protocolo ARP pode ser facilmente lida, por exemplo através do protocolo SNMP.

3.3.1 SNMP ARP/Routing Table Reading

O *Simple Network Management Protocol* (SNMP, RFC 3416 [PRESUHN]) é um protocolo de nível de aplicação componente da pilha TCP/IP (camada 4) utilizado para distribuir informações entre dispositivos da rede. Um dispositivo gerenciável através de SNMP executa um agente, que é o *software* que coleta as informações do dispositivo e gera as bases de informação correspondentes (*Management Information Base*, MIB), que são lidas pelos sistemas de gerenciamento (*Network Management Systems*, NMS). Detalhes sobre o características e o funcionamento do SNMP não fazem parte do escopo do trabalho, e podem ser localizados com mais detalhes em [CISCO].

Para tentar criar um mapa de uma rede, um determinado elemento pode ler a sua própria tabela ARP para criar uma lista de elementos já conhecidos. Com base nessa informação, é possível "perguntar" a cada um deles (através do SNMP) que outros elementos da rede eles conhecem (lendo as suas tabelas ARP). A partir dessa informação, é possível ampliar a lista de elementos conhecidos. Outro uso

Tabela 3.4: Exemplo de ARP via SNMP

RFC1213-MIB::atPhysAddress.2.1.192.168.0.1 = Hex-STRING: 00 02 55 C0 F7 2C
RFC1213-MIB::atPhysAddress.2.1.192.168.0.115 = Hex-STRING: 00 09 6B 98 69 05
RFC1213-MIB::atPhysAddress.2.1.192.168.0.117 = Hex-STRING: 08 00 20 C0 A8 D4
RFC1213-MIB::atPhysAddress.2.1.192.168.0.245 = Hex-STRING: 00 02 55 AA FA 81
RFC1213-MIB::atPhysAddress.2.1.192.168.0.254 = Hex-STRING: 00 02 A5 D9 48 9C

para essa técnica é descobrir as diferentes rotas que possivelmente possam existir na rede, de forma a também criar um mapa de *links* de conexão externos (roteadores, *firewalls*, gateways, etc).

Essa técnica possui diversas desvantagens, dentre as quais podemos citar:

- SNMP não é um protocolo amplamente difundido, e portanto pouco utilizado principalmente em estações de trabalho e servidores, apesar de bastante conhecido e utilizado em roteadores e *switches*.
- Mesmo que o protocolo esteja implementado, diferentes versões e implementações podem possuir características limitadas, que muitas vezes não são adequadas a esse propósito.
- Em outros casos, fabricantes podem implementar as suas MIBs de maneira diferente, tornando muito difícil determinar com precisão o local e formato corretos para a captura dos dados.
- Nenhum *host* conhece a rede como um todo, e muito provavelmente algum *host* não fala com nenhum outro, o que possivelmente irá gerar uma informação incompleta.
- Por mais informação que um *host* conheça, ela é armazenada por um curto espaço de tempo, e portanto pode ser perdida antes que alguém a capture.
- Por fim, similar a outras técnicas já descritas, muitas vezes esse protocolo é bloqueado nos *firewalls*, visto que deixa à amostra diversas informações que muitas vezes não se deseja que sejam divulgadas.

3.3.2 ARP Scan

Apesar de ser possível encontrar vários relatos de ocorrências sobre o *ARP scan*, existe pouca documentação demonstrando seu funcionamento e uso. A melhor explicação encontrada foi na edição de março de 2001 da revista *Connect*, editada pela Novell.

> *"Hackers podem descobrir dispositivos ativos em um segmento da rede local enviando uma série de broadcasts ARP e incrementando o valor do endereço IP de destino em cada pacote. (...) O ARP scan é à prova de falhas: cada dispositivo em uma rede deve responder quando seu endereço IP é mencionado em um broadcast ARP."* [CHAPPEL] (tradução livre)

Assim sendo, essa técnica é bastante similar ao PING Sequencial, com a diferença que as máquinas são obrigadas a responder às solicitações (enquanto no PING, a resposta é opcional), e só funciona se o solicitante estiver no mesmo segmento de rede. Outra vantagem é o fato de ser uma técnica difícil de ser bloqueada, já que não passa por filtragem a não ser na existência de *switches* gerenciáveis que consigam impor limites na quantidade de *broadcasts* por segundo. Contudo, é uma técnica facilmente detectada por *sniffers* e sistemas de detecção de intrusão.

3.4 DNS Table Query

O *Domain Name System* (DNS), definido pelas RFC 1034 [MOCKAPETRIS] e RFC 1035 [MOCKAPETRIS], é o protocolo de camada de aplicação (nível 4 do TCP/IP) que implementa a tradução entre endereços IP e nomes de *hosts* (com ou sem sufixo de domínio). O protocolo foi originalmente criado a fim de simplificar a localização de máquinas na Internet, já que os seres humanos conseguem lidar muito mais facilmente com nomes do que números (principalmente sequências numéricas sem uma lógica precisa).

Tabela 3.5: Cabeçalho DNS

Identificação	QR	Operação	AA	TC	RD	RA	Z	AD	CD	Retorno
Total de Perguntas	Total de Respostas no Registro de Recursos									
Autoridades no RR	Total de Registros de Recursos Adicionais									
Perguntas										
Respostas dos Registros de Recursos										
Autoridades do Registro de Recursos										
Registro de Recursos Adicionais										

O esquema de resolução de nomes consiste basicamente no cliente (máquina que deseja acessar outra máquina) enviando uma solicitação ao servidor DNS (também conhecido como *resolver*). O *resolver* localiza a informação solicitada em sua base de dados de domínios locais (para os quais ele possui autoridade de

responder), não encontrado procura o endereço na sua base de endereços recentemente localizados (*cache*) e, por fim, não encontrando novamente, pergunta a outro servidor superior (normalmente um *root server* global). De posse da resposta, ela é enviada para o solicitante, que por sua vez inicia o procedimento de comunicação desejado. Este processo não leva mais que alguns milisegundos para ser completado.

Uma tabela de DNS (como é comumente chamada a base de dados que armazena as informações de um determinado domínio) pode conter diversas informações sobre um conjunto de *hosts* (outra expresão de uso comum é “zona de DNS”). Normalmente essa tabela armazena diversas relações entre nomes de máquina e endereços IP (registros *A*), além de seus apelidos (registros *CNAME*) e os endereços dos hosts responsáveis por receber e-mails para esse domínio (registros *MX*) ou ajudar a resolver outros endereços (registros *NS*). Muitos administradores também utilizam essas tabelas para armazenar outras informações como descrições das máquinas (a que se destinam, ou qual sistema operacional utilizado), indentificação de roteadores e serviços da rede.

Tabela 3.6: Exemplo de Tabela de DNS

dominio.com.br	SOA	ns1 hostmaster 2004082801 7200 3600 604800 86400
dominio.com.br	NS	ns1.dominio.com.br
dominio.com.br	MX 10	smtp.dominio.com.br
portal	A	200.200.200.200
recursos	A	100.100.100.100
ns1	CNAME	recursos
www	CNAME	portal
smtp	CNAME	recursos

Utilizando as ferramentas apropriadas é possível ler as informações das tabelas de DNS de forma que seja possível construir uma lista das máquinas que estão disponíveis sob a “jurisdição"de um dado domínio. O protocolo DNS, através do item 4.3.5 da RFC 1034 [MOCKAPETRIS] define um procedimento conhecido como "transferência de zona"("*zone transfer*”). Inicialmente a transferência de zona foi criada com o objetivo de possibilitar a replicação das informações de um servidor DNS primário para um ou mais servidores secundários. Cada vez que uma zona é editada, o número serial da mesma é incrementado. Os servidores secundários checam regularmente (em um período de tempo configurável) por tais alterações, que quando detectadas disparam o processo de transferência completa da zona. Esta transferência de zona também pode ser executada por qualquer *host* que não seja um servidor de DNS secundário, desde que possua as ferramentas necessárias.

Uma das grandes vantagens do uso dessa técnica reside no fato de que ela obtém uma resposta no menor tempo possível, já que não é necessário procurar pelas informações na rede. Uma transferência completa de uma zona DNS para uma grande quantidade de máquinas pode durar menos do que uma dezena de milisegundos. Outra grande vantagem é que em algumas situações uma única tabela de DNS pode conter informações sobre hosts de várias redes diferentes, sejam elas fisicamente presentes (outras sub-redes IP rodando na mesma LAN) ou até mesmo geográficamente distribuídas (como as redes de outras filiais espalhadas em uma rede WAN), o que possibilita a criação de um mapa da rede muito mais preciso. Não podemos esquecer também que essas tabelas podem armazenar informações sobre roteadores, facilitando até mesmo a criação de um gráfico de topologia das redes.

Esta técnica, no entanto, sofre de sérios problemas com relação à veracidade e precisão das informações. Não existe nenhum documento que obrigue a publicação de informações em servidores DNS que atendem a redes que não possuem serviços públicos, ou seja, se uma determinada empresa apenas possui acesso à Internet, e não publica nenhum serviço, ela não é obrigada a possuir um servidor DNS para armazenar as informações sobre os *hosts* que compõem a sua rede. Mesmo as redes que possuem serviços públicos, e que portanto utilizam-se de servidores DNS para divulgar seus endereços, também não são obrigadas a cadastrar todas as máquinas existentes. Ainda que uma determinada empresa decida por cadastrar todos os endereços de todas as máquinas, eles podem estar cadastrados em uma tabela de DNS indepentende, de uso privado, e que só poderá ser acessada internamente à rede, dificultando a localização de informações se o interessado estiver localizado externamente.

O problema mais comum enfrentado pelos interessados no uso dessa técnica tem relação com a própria definição da transferência de zonas. Como ela foi originalmente criada para atender à replicação por parte de servidores secundários, muitos servidores proibem a transferência dessas informações para máquinas que não sejam servidores secundários conhecidos. Assim, muitas vezes essas restrições podem estar definidas na própria configuração do servidor DNS, ou até mesmo em regras de *firewall*.

Não podemos também esquecer que as informações contidas nas tabelas de DNS muitas vezes são cadastradas e/ou alteradas através de intervenção humana, ou seja, suscetível a erros de digitação, informação errada, ou até mesmo defasada, já que algumas vezes uma determinada máquina entra ou sai da rede, mas não existe o interesse imediato de cadastrá-la ou removê-la dessas tabelas.

Apesar de todos os problemas aqui apresentados, essa técnica se torna bastante interessante quando utilizada em conjunto com outras técnicas, como o PING, já

que podemos executar diversos “PINGs” na rede para localizar as máquinas existentes, e procurar os seus nomes nas tabelas de DNS. Também podemos localizar, através própria tabela de DNS, outras redes IP que componham o conjunto total de uma rede, de forma que poderemos aplicar qualquer outra técnica nessa nova rede descoberta.

3.5 Traceroute

De acordo com a RFC 2151 [KESSLER/SHEPARD], o *traceroute* é uma ferramenta utilizada para determinar o caminho traçado por um pacote, durante a sua transmissão até o destino final. O processo é iniciado por meio do envio de um pacote UDP para uma porta invalida do *host* de destino com o valor 1 no campo TTL (*Time To Live*), de forma que assim que o pacote alcança o primeiro *gateway* (roteador ou *firewall*) este retorna uma mensagem ICMP *TIME_EXCEEDED* (tempo excedido em trânsito), e é listado como o primeiro *hop* (salto). A seguir é enviado outro pacote com o valor 2 no TTL, que por sua vez ocasiona um *TIME_EXCEEDED* no segundo roteador. O processo é repetido acrescentando-se 1 no campo TTL até que seja retornada uma mensagem ICMP de *Destination Unreachable*, que ocorre quando o pacote alcança o destino final e não consegue conectar-se à porta solicitada (que, por definição, é inválida). Inicialmente o *traceroute* foi criado como uma ferramenta auxiliar na depuração de problemas de roteamento, de forma a facilitar a localização de rotas ou roteadores mal configurados, ou pontos de gargalos no caminho.

Tabela 3.7: Exemplo de *Traceroute*

traceroute to 10.0.200.10 (10.0.200.10), 30 hops max, 38 byte packets
1 10.0.100.1 (10.0.100.1) 0.438 ms 0.371 ms 0.250 ms
2 * * *
3 172.16.0.1 (172.16.0.1) 17.162 ms 27.681 ms 22.212 ms
4 * * *
5 10.0.200.10 (10.0.200.10) 56.839 ms 65.846 ms 84.232 ms

O *traceroute* é mais eficiente quando combinado com outras técnicas, já que para dar início ao processo é necessário determinar um destino a ser traçado. Essa informação pode ser passada manualmente, sabendo-se o que é necessário localizar, ou automaticamente através da listagem de redes disponíveis. Essa listagem também pode ser determinada através de uma verificação na tabela de rotas da máquina de destino, ou através da listagem de endereços disponíveis em uma tabela de DNS.

No caso de inicar-se o processo através da verificação da tabela de rotas, é necessário, antes de tudo, localizar um destino válido que irá auxiliar no processo de *traceroute*. Essa localização pode ser feita através de um *Broadcast PING* ou um *PING* sequencial. De outra forma, o *DNS Table Query* permite capturar uma listagem de máquinas existentes em um determinado domínio de Internet. A segunda técnica (*DNS*)difere da primeira (*PING*) pois a máquina de origem não precisa conhecer todas as redes disponíveis, já que muitas vezes elas podem ser alcançadas através da rota padrão da rede, não exigindo rotas locais.

Um dos problemas encontrados no uso dessa técnica é que muitas empresas configuram os seus roteadores e/ou *firewalls* para não responder ao *traceroute*, o que muitas vezes é feito por motivos de segurança, para esconder os caminhos, e por conseguinte outras redes por onde um pacote passa até alcançar o seu destino. Dessa forma, a técnica torna-se imprecisa até o ponto em que talvez seja possível alcançar o destino final, mas não seja possível determinar com precisão quem são os *gateways* no caminho (apesar de ser possível saber quantos são). Por outro lado existe também a possibilidade do próprio destino não estar preparado para responder, seja através de uma configuração local, ou através de um filtro em um roteador ou em um *firewall*, o que pode fazer com que o *traceroute* demore muito tempo para "entender" que o destino não poderá ser alcançado (esse tempo é determinado através de quantidade máxima de *hops* permitidos.

3.6 Dynamic Routing

O BGP (*Border Gateway Protocol*), conforme definido pela RFC 1771 [REKHTER], é um protocolo classificado como EGP (*Exterior Gateway Protocol*), ou seja, é utilizado para o roteamento "exterior", principalmente na interconexão de meta-redes públicas, também conhecidas como Sistemas Autônomos (AS, *Autonomous Systems*). Cada AS consiste em uma grande rede, gerenciada por uma única empresa, que pode ou não sub-locar sua infra-estrutura para outras redes menores. É o caso das operadoras de telefonia, que fornecem acesso à internet por meio de circuitos especializados conectados ao seu *backbone* (espinha dorsal, ou rede principal). Os protocolos EGP, como o BGP, são utilizados para trocar informações sobre a existência dessas sub-redes, e como alcançá-las, para outros AS, formando uma rede pública maior (como a Internet). A base de funcionamento do BGP é ser um protocolo baseado em *vetor de caminhos*, ou seja, ele encaminha os pacotes através das rotas que passam por uma menor quantidade de interligações.

Esse tipo de roteamento, no entanto, está reservado hoje ao uso por grandes corporações que operam tais sistemas autônomos. São raros os casos em que uma empresa precisa utilizar esse recurso, mesmo em casos onde é necessário interli-

gar uma grande quantidade de redes (como as multi-nacionais). Nos casos mais comuns, ou seja, internamente aos sistemas autônomos (como os *backbones* nacionais) ou até mesmo em grandes empresas, são utilizados protocolos do tipo IGP (*Interior Gateway Protocol*), que tratam do roteamento "interior". Os mais utilizados são o RIP e o OSPF.

O OSPF (*Open Shortest Path First*) é definido pela RFC 2328 [MOY] como sendo um protocolo de roteamento baseado em *estado do link*, ou seja, todas as rotas são constantemente propagadas, juntamente com seus estados (limitando-se a domínios pré-configurados). Inicialmente o OSPF sempre escolhe a rota de menor custo (daí o seu nome), contudo se no meio dessa rota houverem outras rotas inoperantes, ela nem sequer será cogitada, e o roteamento será direcionado para a de segundo menor custo.

Outro protocolo do tipo IGP é o RIP (*Routing Internet Protocol*), definido pela RFC 2453 [MALKIN] como um protocolo de *vetor de distância*, ou seja, sempre escolhe a rota de menor custo, não importando a carga nem o estado dos circuitos. Esse é o protocolo mais utilizado hoje em dia, mas tem sido substituído pelo OSPF por vários problemas, principalmente o fato de não analisar os estados, bem como limitações como o registro máximo de 15 saltos por rota e a falha no tratamento de *loops* (ou seja, rotas que seguem vários caminhos mas que acabam retornando ao ponto de origem).

Todos os protocolos mencionados baseiam-se na premissa de que suas informações são regularmente disponibilizadas publicamente (como no caso do RIP e do BGP) ou replicadas entre conjuntos de roteadores distintos (como é o caso do OSPF). Independente da situação, é possível utilizar-se dessas informações para construir um mapa dos caminhos disponíveis para tráfego de dados, tanto interno quanto externamente. Essas informações também podem ser utilizadas no sentido de localizar outras redes inicialmente desconhecidas.

Durante a pesquisa realizada para este trabalho, poucas foram as referências encontradas a esta técnica, e muito pouco se descreveu sobre seus processos. Também não foi possível encontrar nenhuma ferramenta que pudesse elucidar melhor o seu funcionamento. Contudo, é importante citá-la como uma possibilidade para análise futura.

3.7 Passive Discovery

A técnica de localização de elementos através de **descoberta passiva** é aquela que envolve principalmente a utilização de *packet sniffers*, que são tipos de programas especiais que permitem a visualização de todo o tráfego de uma rede. Estes programas normalmente são utilizados por pessoas mal-intencionadas pois possi-

bilitam a captura das informações que trafegam na rede e podem servir para coletar senhas, dados confidenciais, mensagens instantâneas ou de e-mail, etc. Contudo esse mesmo tipo de programa pode ser muito útil para a localização de elementos que não estejam respondendo a nenhuma das técnicas mencionadas anteriormente.

Os *sniffers* tiram proveito da premissa básica do protocolo *Ethernet* de que todo o tráfego enviado na rede é transmitido em *broadcast*, ou seja, enviado para todas as máquinas presentes em um barramento, onde cada uma delas descarta os pacotes que não lhe pertencem. No caso de máquinas que executam os *sniffers*, as interfaces de rede são colocadas em modo promíscuo, ou seja, todo o tráfego passado na rede é aceito, mesmo que não seja destinado para essa máquina.

Utilizando-se das ferramentas certas, é possível capturar todo o tráfego da rede, ignorando os dados, e armazenando apenas os endereços de origem e destino dos pacotes. De posse dessas informações, é possível construir um banco de dados de todas as máquinas ativas na rede em um dado momento. Além disso, comparando os endereços IP com os endereços MAC, é possível até mesmo saber quando mais de um endereço IP pertence à mesma máquina, ou quando ela muda o seu endereço após um tempo.

A grande desvantagem dessa técnica é que, pelo fato de ser passiva, depende essencialmente da existência de tráfego entre as máquinas. Dessa forma, se a localização é executada em um curto período de tempo, ou em períodos espaçados, é possível que alguns elementos não sejam detectados por não estarem ativos naquele momento, ou até mesmo por não estarem gerando nem recebendo tráfego da rede. Outra desvantagem é que além de ser possível capturar os endereços locais, também serão capturados endereços externos à rede, quando essa comunica-se com outras, o que gera um trabalho extra de filtragem, que também pode ser impreciso. Essa filtragem poderá ser imprecisa caso todas as faixas de endereços locais não sejam conhecidos de antemão. Por fim, mesmo que o *sniffer* seja executado initerruptamente, pode-se levar várias horas, ou até mesmo dias, para detectar todas as máquinas de uma rede. Dessa forma, essa técnica pode ser utilizada como complementar, para localizar elementos que não tenham sido localizados pelas técnicas anteriores.

Uma observação importante a ser mencionada é o fato de que muitos administradores de redes pensam que o uso de *sniffers* só é possível em redes dotadas de *hubs*, que facilitam o uso dessas ferramentas já que o tráfego repassado por eles obrigatóriamente é replicado para todas as estações. No entanto existem outras técnicas acessórias que possibilitam o uso de *sniffers* mesmo em redes dotadas de *switches* (como o *ARP spoof* e o *MAC flood*), que forçam outras máquinas (ou até mesmo o *switch*) a encaminhar seus pacotes também para a máquina onde o *sniffer* está sendo executado. Também é importante citar que existem ferramen-

tas específicas para detectar o comportamento dos *sniffers*, tanto pelos IDS como ferramentas exclusivas para esse fim.

3.8 Comparativo entre as técnicas

A partir do estudo geral das técnicas hora apresentadas neste trabalho, é possível perceber que nenhuma das técnicas encerra em si todas as características e recursos necessários a uma localização completa dos elementos de uma ou mais redes. É possível que algumas delas alcance uma grande proporção de acertos, e outras talvez necessitem de técnicas complementares para concluir o seu trabalho. De uma maneira geral a escolha da técnica correta depende de vários fatores, principalmente das características da rede e a disponibilidade de ferramentas.

Como forma de auxiliar nessa decisão, é possível criar um quadro comparativo entre as técnicas (Tabela 3.8), possibilitando uma visão geral e uma escolha mais adequada. Os itens analisados são:

Abrangência:
A técnica pode ser aplicada a *Domain*, *Backbone* ou *Ambos*.

Posição:
Se para aplicar a técnica o usuário precisa estar *Local* ou *Remoto* em relação à rede, ou *Ambos* quando a posição indifere (podendo ou não influenciar o resultado).

Ferramentas:
Se quantidade e facilidade de localizar ferramentas é *Alta*, *Média* ou *Baixa*. Vale resaltar que essa análise é baseada em ferramentas livres na plataforma *Linux*. Os níveis de facilidade variam de acordo com a disponibilidade prévia de ferramentas na própria distribuição, existência de pacotes para transferência na Internet, ou necessidade de compilação. Além disso, conta também a quantidade de ferramentas disponíveis.

Precisão:
Baseado nas características da própria técnica, bem como dos protocolos envolvidos, análise de comportamentos relacionados a segurança (bloqueio por *firewalls*, configurações dos Sistemas Operacionais, etc) pode-se determinar, por aproximação, o nível de precisão que pode ser alcançado pela técnica. Deve-se notar que essa precisão varia também de acordo com a informação que a técnica se propõe a localizar.

Com base na tabela acima, é possível verificar quais seriam as melhores ferramentas para determinadas situações.

Tabela 3.8: Visão geral de características e recursos

Técnica	Abrangência	Posição	Ferramentas	Precisão
PING Sequencial	Ambos	Ambos	Alta	Alta/Média
broadcast PING	Domain	Ambos	Alta	Baixa
TCP/UDP Scan	Ambos	Ambos	Média	Média
SNMP Routing Reading	*Backbone*	Ambos	Baixa	Média
SNMP ARP Reading	Domain	Ambos	Baixa	Baixa
ARP Scan	Domain	Local	Baixa	Alta
DNS Table Query	Domain	Local	Média	Baixa
Traceroute	*Backbone*	Ambos	Alta	Baixa
Dynamic Routing	*Backbone*	Ambos	Baixa	Baixa
Passive Discovery	Domain	Local	Média	Alta/Média

3.8.1 Domain Topology Discovery

As técnicas de *PING Sequencial* e *ARP Scan* são as mais precisas para este tipo de localização. A primeira possui uma maior facilidade de acesso a ferramentas, já a segunda exige um pouco mais de trabalho para conseguir a boas ferramentas e aprender a utilizá-las. Contudo os resultados obtidos a partir de qualquer uma dessas técnicas é o maior encontrado quando aplicado a redes locais, principalmente pelo fato de não sofrerem filtragem de pacotes, principalmente no caso do *ARP Scan*. O único problema com o *PING Sequencial* é com relação ao uso de *firewalls pessoais*, que tem se difundido bastante hoje e podem ser um impecilho, mas enquanto o seu uso não se popularizar, essa técnica continuará rendendo ótimos frutos.

É importante também fazer uma menção honrosa ao *Passive Discovery*. A quantidade de ferramentas de *sniffer* tem aumentado continuamente, principalmente por serem muito úteis para o diagnóstico de problemas em redes. Com um pouco de paciência do usuário, e com as ferramentas corretas para filtrar, classificar e armazenar os dados coletados, a precisão aumenta proporcionalmente ao tempo em que a técnica é aplicada. Isso significa dizer que por quanto mais tempo dados são coletados, maior a precisão e o volume de informações que pode ser auferido pela rede.

Das técnicas que podem ser aplicadas a localização em redes locais, o *Broadcast PING* apresentou-se menos interessante. O principal fator que contribui para o seu baixo rendimento é o fato de que muitos sistemas operacionais e equipamentos de rede não respondem mais aos pacotes de *broadcast* a nível de IP (mas continuam respondendo no ARP). Dessa forma, é possível localizar alguns elementos, mas não existe nenhuma garantia de que o volume localizado é proporcional ou

correspondente ao número real de elementos na rede.

3.8.2 Backbone Topology Discovery

A técnica de *SNMP Routing Table Reading* é a que apresenta o maior nível de precisão para esse ambiente. A excessão seria em redes públicas onde os roteadores não permitem acesso às suas tabelas SNMP. Quando estas estão disponíveis, as informações podem ser facilmente capturadas e um mapa muito aproximado da realidade pode ser criado.

As técnicas mistas de *PING Sequencial* e *TCP/UDP Scan* podem apresentar também bons níveis de precisão, mas sofrem de problemas com relação à filtragem dos pacotes, que podem ser feitos nos roteadores ou nos *firewalls*.

Entretanto a técnica mais promissora para essa situação é o *Dynamic Routing Reading*, que pode auxiliar na criação de mapas muito mais precisos das redes, já que o comportamento dos protocolos de roteamento dinâmico permite coletar muito mais dados que qualquer outra técnica. Contudo seu uso é muito restrito pois poucas redes utilizam essses protocolos.

Com relação ao *Traceroute*, seu uso é muito limitado, portanto a pior técnica a ser aplicada, já que considera apenas os caminhos possíveis entre a origem e o destino. Dessa forma, se não houver destinos conhecidos, não há como traçar uma rota. Por outro lado, as rotas podem variar de acordo com o destino ou até mesmo com a própria origem. Isso significa dizer que se não for possível mudar o ponto de vista, os dados coletados serão mínimos e insuficientes para construir um mapa ao menos aproximado.

Capítulo 4

Análise de Ferramentas

A análise das ferramentas foi feita utilizando a plataforma GNU/Linux, distribuição Conectiva, versão 10. Todas as ferramentas selecionadas são liberadas sob a licença GPL (*General Public License* [GNU]) ou similar, e portanto de livre acesso, uso e distribuição gratuita. A escolha pela plataforma e o tipo de ferramentas baseou-se principalmente na disponibilidade, facilidade de aquisicão (transferência via Internet) e na inexistência de custos, que poderiam inviabilizar o trabalho. Além disso, as plataformas livres têm se demonstrado a principal escolha dos gerentes de redes e consultores independentes, tanto pelos motivos supracitados como pela grande disponibilidade de ferramentas para todos os fins, bem como facilidade de criação de novas ferramentas e customização das existentes.

A escolha das ferramentas foi feita através de buscas no *Freshmeat* [OSTG], que tem se demonstrado como a principal fonte de referência e informação sobre *softwares* disponíveis para plataformas livres. Alguns desses *softwares* também podem ser encontrados no *SourceForge* [OSTG], que é o recurso mais utilizado pelos desenvolvedores dessas plataformas para hospedar seus arquivos, sites e listas de discussão. Também foram levados em conta a pré-disponibilidade das ferramentas na própria distribuição de testes, bem como a facilidade de capturá-las e integrá-las ao ambiente.

A rede onde foram executados os testes compreende aproximadamente 150 *hosts*, entre servidores, estações e roteadores, com uma boa diversidade de sistemas operecionais (Linux, Solaris e Windows, em várias versões) e tipos de equipamentos (*hubs*, *switches*, roteadores, impressoras em rede e multi-funcionais). A gerência da empresa autorizou a aplicação das ferramentas para testes, mas o uso de imagens ficou restrito à uma pequena rede com acesso restrito. Também não foi autorizada a divulgação de informações detalhadas sobre os resultados as quais possam causar qualquer tipo de comprometimento ou vazamento de informações

sigilosas.

4.1 Nmap

O Nmap [INSECURE] é uma das ferramentas relacionadas a segurança e gerenciamento de redes mais difundidas no ambiente Linux. Ela foi inicialmente criada com o objetivo de detectar portas abertas em um destino, mas também pode ser utilizada para a localização de dispositivos em um rede, ou para a verificação de alguns itens básicos de segurança, como portas abertas em um *host* (servidor ou estação), verificação de regras de *firewall* e testes de sistemas IDS (*Intrusion Detection Systems*), ou até mesmo detectar qual sistema operacional ou que tipo de equipamento se trata um determinado *host*. Essa ferramenta é baseada principalmente nos conceitos de *network scan* já que faz buscas simples através de tentativa e erro, e não possui meios simples de documentar os resultados (apenas armazenamento em *log*) ou validá-los. Talvez estas sejam a grande vantagem e o principal motivo por ter se tornado tão popular: simplicidade, velocidade e praticidade.

O Nmap é dotado de várias técnicas, algumas delas bem documentadas e que se aproveitam das próprias caracterísitcas do protocolo TCP/IP, bem como outras um tanto quanto obscuras pois se aproveitam de falhas de implementação. Das técnicas descritas neste trabalho, o Nmap usa as seguintes:

PING Sequencial

O Nmap recebe uma lista de endereços e envia uma requisição de *ICMP echo* para cada um deles, ao receber a resposta é exibido o endereço com uma mensagem de que ele está ativo. A cada endereço exibido é executado também uma busca reversa de DNS, de forma a poder informar o nome da máquina, contudo às vezes isso não é possível (e o endereço é mostrado numericamente), ou não é desejado (e utiliza-se um parâmetro para desativá-lo).

Conforme discutido anteriormente, alguns elementos de rede são configurados para não responder às requisições de PING, ou não podem por estarem atrás de um *firewall* que não o permite. Já prevendo tal situação, o Nmap também envia uma solicitação de conexão na porta 80 do protocolo TCP, e havendo resposta o elemento é considerado ativo. Nem sempre ela estará aberta, já que se trata da porta do serviço de *web* (WWW), mas tal porta é considerada pois muitas vezes os *firewalls* permitem a conexão, já que é uma porta normalmente utilizada por um serviço público. Por outro lado, existem também alguns equipamentos como *switches* ou roteadores que podem ser gerenciados através de um servidor *web* interno, portanto possuem a porta 80 ativa.

Duas observações são importantes. Primeiro que o PING também é executado em qualquer outro tipo de *scan* que se faça com o Nmap. Na verdade o Nmap

Figura 4.1: PING Sequencial com Nmap

só executa o *scan* se a máquina responder ao PING. Este comportamento também pode ser desabilitado através de parâmetros. Segundo, quando vários endereços de destino são especificados (uma faixa, ou uma subrede) o PING é executado sequencialmente, contudo muitas vezes este comportamento não é desejado, já que os sistemas IDS conseguem detectar e alarmar essa situação. Para isso existe uma opção para que os PINGs sejam executados em ordem aleatória, bem como em intervalos de tempo distintos.

No exemplo acima, pode-se verificar que uma faixa inteira de endereços foi verificada (254 possíveis máquinas), e foram obtidas 5 (cinco) respostas. Na realidade essa rede consiste em 6 (seis) máquinas, porém uma delas é um *firewall* e está configurada para não responder ao PING. O parâmetro -n"desabilita a verificação reversa dos endereços.

TCP/UDP Scan

O comportamento padrão do Nmap é procurar por todas as portas TCP abertas no destino especificado. Para tal ele tenta conectar em cada uma das portas desde a 1 até a 65535 (valor máximo de uma porta do TCP/IP), ou em uma faixa de portas especificada. Essas conexões podem também ser tentadas utilizando o protocolo UDP. Independente do protocolo a ser testado, o Nmap sempre testa primeiro se o destino está operacional enviando um PING antes, e só executando o *scan* se o mesmo responder. Contudo, esse comportamento também pode ser alterado para que o teste não ocorra e o *scan* seja feito independente da situação.

Considerando que este tipo de *scan* é facilmente detectado pelos IDS (como discutido anteriormente), o Nmap disponibiliza outros tipos, sendo o principal o *SYN Scan*. Esta técnica consiste em uma conexão incompleta em uma porta, ou seja, envia-se a solicitação conexão (*SYN*), e recebe-se a permissão (*ACK*), mas logo em seguida envia-se um cancelamento (*RST*), o que faz com que a conexão

Figura 4.2: *SYN Scan* com Nmap

seja detectada pelo Nmap (já que o destino respondeu positivamente), mas não é enviada a confirmação que concluia conexão (*ACK*), de forma que o destino ignora a tentativa. A vantagem desta técnica é que não é registrada a tentativa de conexão. No entando, isso não impede que sistemas IDS ainda detectem o comportamento.

No exemplo acima a busca foi limitada às primeiras 512 portas TCP a fim de obter-se um resultado menos extenso. Note também que uma das máquinas não possui nenhuma porta aberta, mas, como demonstrado no exemplo anterior ela responde ao PING. O tempo reduzido decorre de ter-se executado o *scan* apenas nas máquinas ativas (que respondem ao PING), caso o PING fosse habilitado, esse procedimento poderia levar várias horas (já que a busca ocorreria nos 254 endereços válidos).

List Scan

Essa técnica usada pelo Nmap na verdade é uma variação do **DNS Table Listing** discutido anteriormente. A diferença é que o Nmap apenas lista todos os endereços IP da(s) faixa(s) solicitada(s), e faz uma busca reversa pelo nome de cada um deles (DNS reverso) sem executar qualquer tipo de *scan* (seja PING, ou

busca por portas).

Não foi possível a demonstração de um exemplo pois a rede utilizada para testes durante a elaboração deste trabalho não possui DNS reverso.

4.2 KNetmap

O KNetmap [CORBIN] trata-se basicamente de uma interface gráfica para o Nmap. Apesar de depender totalmente das funcionalidades fornecidas por esta ferramenta, a interface do KNetmap facilita a visualização das informações, e agiliza o seu uso por possuir menus e vários itens de configuração bastante intuitivos, de forma que marcar ou desmarcar uma opção ativa ou desativa uma determinada característica do Nmap.

Utilizando as opções padrão, o KNetmap executa uma varredura do tipo ***SYN Scan***, com verificação de versão dos serviços (conecta, tenta negociar comunicação, e verifica a versão do *software* que está executando na porta), além de verificação do sistema operacional. Este tipo de checagem é bastante intrusiva e chama muita atenção, pois além de executar PINGs para verificar quais máquinas estão ativas, também conecta em cada uma das portas, fazendo com que a máquina de destino receba uma certa carga de conexões, mesmo que mínima. Além disso, o processo todo torna-se muito demorado, já que todas as verificações são feitas em um único *host* antes de passar para o seguinte. Nos testes executados, a verificação de uma estação Windows levou aproximadamente 10 minutos. A verificação completa de uma rede classe C (254 IPs válidos), com 150 máquinas ativas, levou cerca de 4 horas.

A grande vantagem do uso dessa ferramenta é o resultado final. O KNetmap cria uma árvore compreensiva onde são listados os componentes de cada rede detectada (se mais de uma). Ao selecionar um dos elementos, tem-se acesso a uma janela com as informações completas sobre o mesmo como as portas abertas, que *software* estão sendo executados nessas portas, e qual o sistema operacional executado (ou tipo de equipamento, no caso de roteadores, *switches*, etc). Outra caracterísitca interessante, porém de uso mais restrito, é a capacidade de ler e gravar arquivos de log do Nmap em formato XML (através do parâmetro "-oX arquivo.xml"). Com isso é possível executar o Nmap em qualquer outra máquina, e analisar os seus resultados através da interface do KNetmap.

Apesar de suas funcionalidades e da interface amigável, o desenvolvimento do KNetmap está congelado desde novembro de 2003, e aparentemente não há nenhuma movimentação para dar continuidade ao projeto. Além disso, não existe documentação disponível, forçando o usuário interessado a procurar mais informações sobre o Nmap do que sobre o KNetmap em si. Mesmo assim, a ferramenta

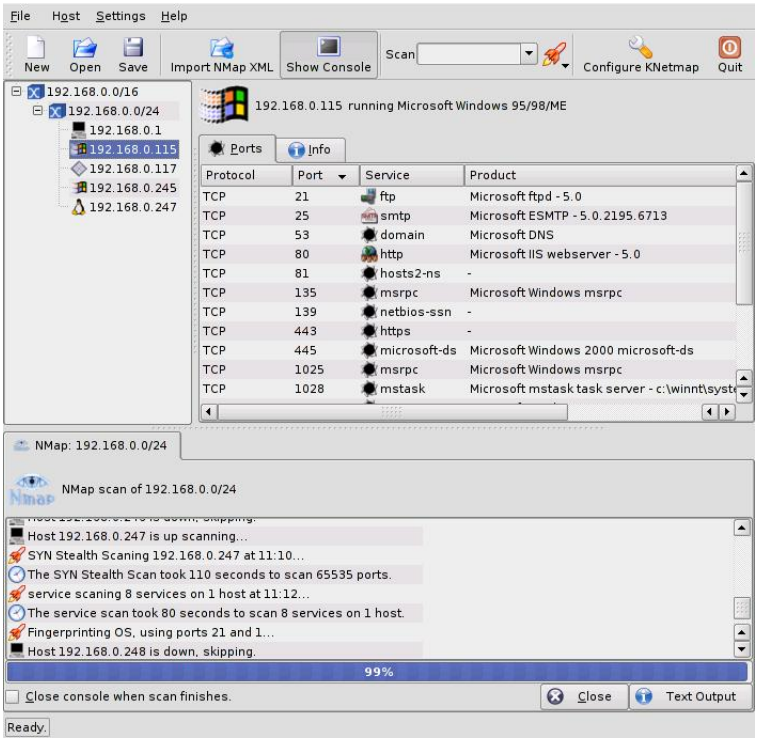

Figura 4.3: Tela do KNetmap

é bastante útil, flexível e amigável.

4.3 FPING

O FPING [DZUBIN] é uma ferramenta que atualmente tem se disseminado com grande velocidade entre os usuários do ambiente Linux, por se tratar de uma maneira simples de enviar PINGs para várias máquinas ao mesmo tempo. Esse programa funciona recebendo uma lista de endereços IP (não aceita faixas, como o endereço de uma rede com máscara, apenas listas de *hosts* individuais) e enviando um ou mais pacotes PING para cada uma deles em paralelo, exibindo os resultados de quais estão ativos.

É possível também exibir diversas estatísticas, como o tempo de resposta e a quantidade de pacotes perdidos para cada destino, bem como quantos endereços responderam, quantos estão inalcancáveis, e até mesmo uma média dos seus tempos de resposta.

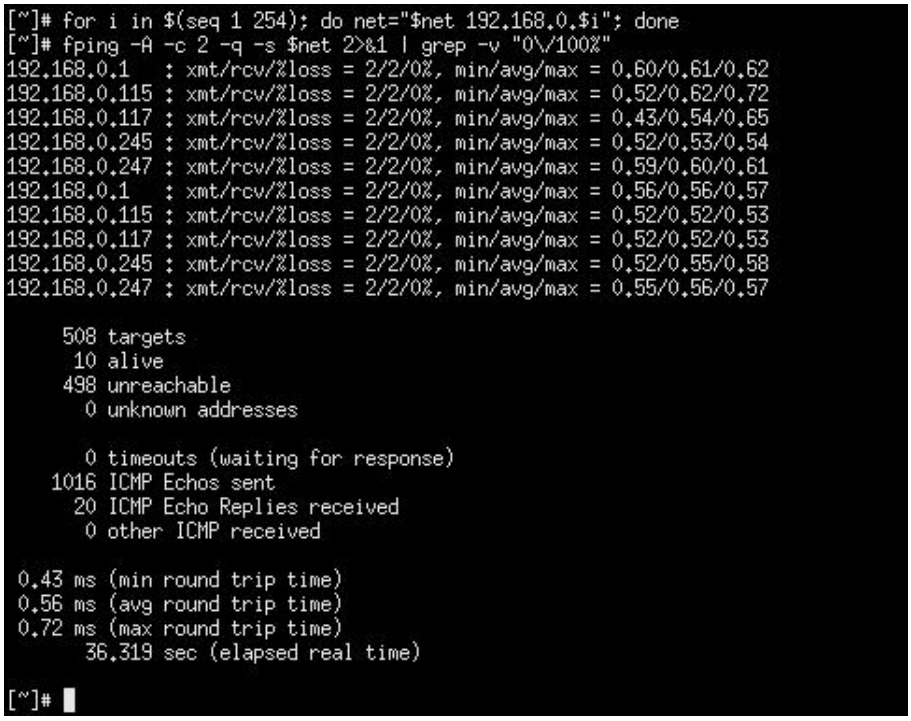

Figura 4.4: Uso do FPING

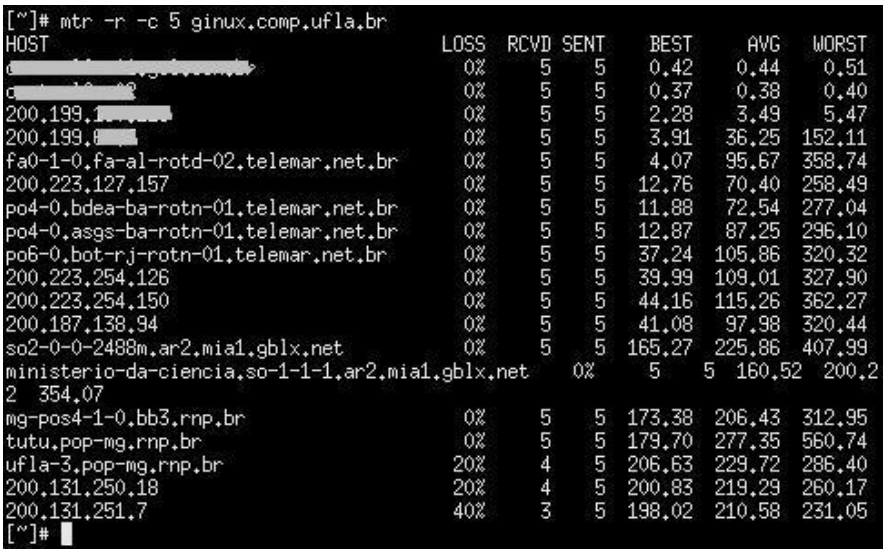

Figura 4.5: Uso do MTR

4.4 MTR

MTR [BITWIZARD] é uma ferramenta que integra as funcionalidades do ***traceroute*** e do PING em uma única interface. O programa recebe como parâmetro um endereço IP de destino, e gera uma tela com a lista de todos os *hops* (roteadores e/ou *gateways*) encontrados no caminho até alcançar o destino. A lista gerada é constantemente atualizada com os tempos de resposta de vários PINGs enviados para cada um dos pontos de interconexão no caminho.

4.5 Cheops

O Cheops [SPENCER] é uma ferramenta criada para o mapeamento e monitoramento de redes, que funciona unicamente através do descobrimento automático de elementos, não havendo a possibilidade de criação manual de itens no gráfico. O projeto pareceu promissor e com uma idéia bastante plausível, mas o desenvolvimento parou em setembro de 2001, quando o desenvolvedor recebeu uma proposta de emprego.

Logo quanto é iniciado, o Cheops pergunta se o usuário deseja fazer uma busca por *hosts* na rede local, que quando aceito dispara um ***broadcast PING*** contra a rede local (auto detectada a partir do endereço IP primário da máquina local).

Conforme relatado anteriormente, tal procedimento é um tanto quanto impreciso, e os resultados comprovam isso. Em uma rede com 150 *hosts* ativos, apenas 15 foram corretamente detectados. Vale resaltar que 7 deles estavam com o protocolo SNMP ativado, o que pode ter facilitado a sua localização, considerando que eram as únicas nessa situação.

Utilizando o resultado do PING é formado um mapa inicial com as máquinas detectadas, e para cada endereço IP é executado um DNS reverso para localizar os nomes. Em seguida é executada a detecção do sistema operacional utilizando o *"QueSO"*, através da técnica de *IP fingerprint* (análise dos cabeçalhos IP), e com isso os ícones genéricos que representam as máquinas são substituídos por outros que representam o sistema operacional ou sua finalidade (como um roteador, por exemplo). Além disso, é feito também um ***SYN scan*** para localizar quais serviços estão ativos na máquina, de forma a facilitar o entendimento do seu papel na rede. Opcionalmente, o usuário pode selecionar a opção de mapeamento, quando são desenhadas linhas interligando as máquinas, gerando um mapa da rede, que é detectado através de uma série de pacotes UDP e ICMP (*traceroute*).

Selecionando-se a opção de adicionar novas redes, o Cheops inicia o processo de localização de máquinas novamente, e estabelece a relação entre a rede previamente localizada e a nova. Essa relação é expressa através da ligação entre os *gateways* (que podem ser roteadores e/ou *firewalls*). Contudo, quanto mais estações e/ou redes são localizadas, mais confuso fica o mapa, já que os ícones não são arrumados automaticamente, necessitando intervenção do usuário.

De uma maneira geral a ferramenta demonstrou-se excelente no sentido de possibilitar uma visão geral e um acesso rápido a estações da rede. Contudo, os seus métodos de localização e a impossiblidade de adicionar ou remover manualmente novos elementos torna o seu uso bastante limitado, o que é bastante compensado pelas facilidades adicionadas pela interface e o sistema de monitoramento.

É possível, por exemplo, clicar com o botão direito sobre um elemento da rede e selecionar a execução direta de comandos como *port scan* (com detecção de serviços e sem o uso do Nmap), *telnet*, *traceroute*, conexão remota através do VNC e muitos outros comandos que podem ser personalizados pelo usuário. Vale a observação de que o *port scan* dessa ferramenta demonstrou-se muito mais rápido do que o Nmap.

Um dos grandes recursos disponibilizados pelo Cheops é o monitoramento de máquinas, que pode ser feito através de PINGs e conexão aos principais serviços como DNS, FTP, SMTP e HTTP. A frequência é configurável de maneira independente, e a falha na resposta de um deles pode ser classificado como um mero aviso (*"warning"*) ou uma condição crítica (*"critical"*). Essas situações podem ser informadas ao administrador através de e-mail, ou registradas em um log no disco

local.

Apesar do desenvolvimento congelado, a ferramenta é muito útil, interessante, elucidativa, e pode ser usada como um recurso rápido para localização de alguns elementos da rede, ou estabelecer a ligação entre eles. Certamente é uma ferramenta que merece mais atenção e quem sabe a criação de uma nova comunidade para continuar o seu desenvolvimento.

Pensando nisso, uma outra equipe decidiu refazer a ferramenta do zero, e assim surgiu o Cheops-ng [PRIDDY]. O sufixo "ng" foi agregado com o intuito de identificar uma nova geração da ferramenta (*new generation*). A intenção é reescrevê-la de tal forma que possa ter mais recursos, mais precisão, e mais facilidade. A grande diferença é que a nova versão funciona com a idéia de *agentes remotos*, ou seja, uma aplicação servidora é instalara em qualquer máquina da rede, e pode executar suas tarefas totalmente comandada a partir de uma interface gráfica em outra máquina qualquer. Isso possibilita o uso remoto do Cheops em redes onde o usuário não está presente. É importante notar que essa conexão não é segura, pois não possui qualquer mecanismo de encriptação nem autenticação.

Outro detalhe verificado nessa nova versão é que o seu desenvolvimento, apesar de constante, ainda está em um estágio bastante inicial. Com poucos recursos, itens importantes como o monitoramento ativo, ainda faltam ser implementados. Além disso, não existe detecção inicial automática, forçando o usuário a informar qual rede deseja procurar. Também foi notado nos testes que a sua detecção ainda é falha, pois só foi possível localizar metade dos *hosts* encontrados pelo Cheops original (nem todos usando SNMP, conforme relatado anteriormente). Outro item que apresentou problemas foi a detecção de sistema operacional, que só funcionou em metade das máquinas localizadas.

Com todos os problemas observados, o Cheops-ng poderá ser uma alternativa no futuro, já que ele pretente ser melhor, mas no momento, mesmo limitado, o Cheops original ainda é a melhor alternativa. Por outro lado, como a nova versão continua GPL, o desenvolvimento tende a ser rápido e pode realmente alcançar os seus objetivos, bastanto receber mais ajuda da comunidade.

4.6 OpenNMS

O OpenNMS [OPENNMS] é uma ferramenta que foi criada com o intuito de atender a todas as necessidades de gerenciamento de redes, baseado no conceito FCAPS [FUTURESOFT] padronizado pela ISO (*"fault, configuration, accounting, performance and security"*, ou em português, *"falha, configuração, contabilidade, performance e segurança"*). O produto está disponível em uma versão gratuita e outra comercial (com suporte especializado). O OpenNMS é baseado

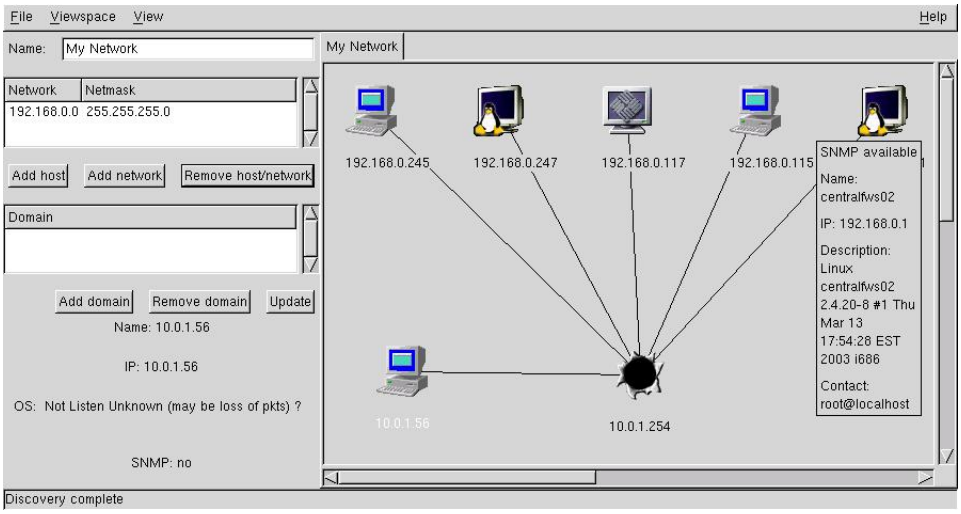

Figura 4.6: Tela do Chops

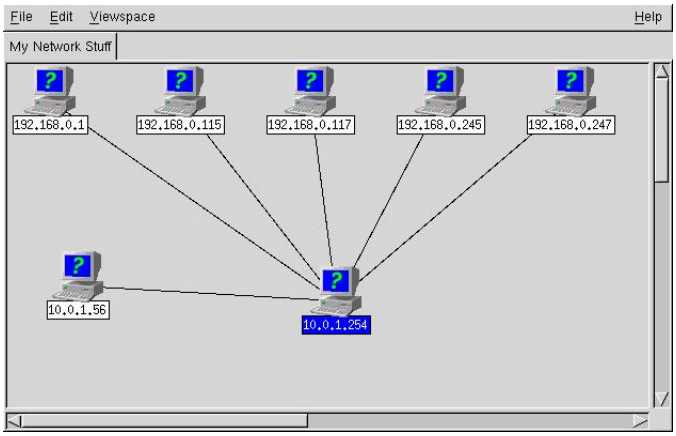

Figura 4.7: Tela do Chops NG

em três pilares: checagem de serviços (disponibilidade), coleta de dados (informações de rede) além de notificação e gerenciamento de eventos. De certa forma, ele é muito similar ao NAGIOS, mas como informado no próprio site, a intenção não é competir com este, mas sim apenas criar uma interface mais simples além de complementar recursos não existentes.

O NAGIOS [GALSTAD] é a ferramenta de monitoramento mais conhecida e utilizada pelos administradores de rede que utilizam Linux. Assim como o OpenNMS, ele é utilizado para monitoramento de disponibilidade (serviços, máquinas e *links*), recursos, notificações automáticas, etc. A equipe do OpenNMS declara que a grande diferença entre os dois projetos é que este é desenvolvido com o intuito de ser uma ferramenta para uso em corporações de grande porte, pois implementa recursos de monitoramento distribuído (através de agentes remotos) e via SNMP que o NAGIOS ainda não é capaz de fazer, a não ser através de *plugins* não-oficiais. Soma-se a isso a declaração na própria documentação original do NAGIOS que ele *"não foi desenvolvido para ser um substituto para uma aplicação de gerenciamento SNMP completa como o HP OpenView ou o OpenNMS"*, já que seus recursos nessa área são limitados.

No contexto deste trabalho, o OpenNMS atende no sentido de possuir um módulo de *network discovery*, criado com a intenção de simplificar a configuração dos elementos que serão monitorados pelo sistema. O processo de localização de elementos é todo baseado no *PING sequencial*, que é enviado para várias faixas de endereços que precisam ser previamente configurados. Esse detalhe (configuração antecipada) é importante, visto que a ferramenta não procura detectar nenhuma rede existente quando inicializada, sendo necessário especificar cada uma desejada antes de ser iniciado o processo de localização. Este processo, apesar de originalmente lento, pode ser acelerado através da criação de vários PINGs em paralelo (*threads*). Os PINGs são executados através do *icmpd*.

Após um elemento ser descoberto através do PING, é acionado o *capsd* (*capabilities daemon*), um processo interno do OpenNMS utilizado para localizar os serviços disponíveis no destino, e que serão monitorados pelo sistema. Os protocolos que podem ser localizados vão desde os serviços convencionais de Internet (DNS, FTP, HTTP, etc) até outros específicos de redes locais (SMB, Citrix, etc).

Como citado anteriormente, o OpenNMS foca principalmente no gerenciamento de recursos e disponibilidade, e por esse motivo sua ferramenta de *discovery* é bastante simples, mas em função da grande complexidade do sistema como um todo (principalmente pelo excesso de dependências e a grande dificuldade de configuração do Tomcat como servidor JSP), não houve interesse em instalá-lo para testes em laboratório. Apesar disso, essa ferramenta demonstra-se bastante completa, e com certeza merece uma atenção especial dos administradores de rede.

4.7 Comparativo entre as ferramentas

Algumas ferramentas são específicas para a aplicação de determinadas técnicas, outras utilizam-se de várias tecnicas em conjunto para alcançar os seus objetivos. Em todas elas o estado de maturidade (estágio de desenvolvimento), e a qualidade da *interface* (gráfica ou não) varia muito, e podem ser decisivos na hora da escolha. A comparação entre as ferramentas (Tabela 4.7) pode ser baseada nos seguintes fatores:

***Interface*:**
Disponibilidade de *interface Gráfica*, *Texto* ou *Ambos* para interação com o usuário.

Disponibilidade:
Facilidade de localização, instalação e uso, medido em *Alta*, *Média* ou *Baixa*. Na dificuldade de uso foram levados em conta itens como opções de configuração, possibilidade de alterar o modo de funcionamento e facilidade de visualização das informações.

Técnicas:
Quantidade de técnicas utilizadas pela ferramenta (conforme analisadas neste trabalho).

Maturidade:
Para análise do nível de maturidade (*Alta*, *Média* ou *Baixa*) leva-se em conta a comparação entre os resultados prometidos (pela documentação ou *site* oficial) e os obtidos. Considera-se também o estágio declarado de desenvolvimento (pela equipe de desenvolvedores) e a data da última atualização.

Armazenamento:
Capacidade própria de armazenar as informações em arquivos ou bancos de dados para consulta regular e/ou futura sem o uso de ferramentas ou técnicas externas (*Sim* ou *Não*).

Ciclo:
Para verificação atualizada das informações se a ferramenta necessita execução *Manual*, *Contínua* ou *Configurável*.

Precisão:
Qualidade e quantidade de informações exibidas em comparação com as informações reais (*Alta*, *Média* ou *Baixa*).

Tabela 4.1: Visão geral de características e recursos

Ferramenta	***Interface***	**Disp.**	**Técnicas**	**Mat.**	**Armaz.**	**Ciclo**	**Precisão**
Nmap	Ambos	Alta	3	Alta	Sim	Manual	Alta
KNetmap	Gráfica	Baixa	2	Média	Sim	Manual	Alta
FPING	Texto	Alta	1	Alta	Não	Config.	Alta
MTR	Ambos	Média	2	Alta	Não	Cont.	Alta
Cheops	Gráfica	Baixa	3	Baixa	Sim	Config.	Baixa
OpenNMS	Gráfica	Baixa	5	Alta	Sim	Config.	Alta

Analisando o quadro comparativo, fica claro que o *OpenNMS* e o *Nmap* são as ferramentas mais adequada para o *network discovery*. Contudo, a dificuldade de implantação do OpenNMS e sua quantidade de recursos o tornam desinteressante quando a intenção é apenas localização, já que essa é uma ferramenta completa de gerenciamento de redes.

O Nmap recebe o mérito de ser a ferramenta mais importante entre todas as analisadas. A única desvantagem do Nmap é a sua impossibilidade de criar gráficos e a dificuldade de condensar as informações coletadas, o que pode ser facilmente resolvido através do uso em conjunto com o KNetmap. Esta ferramenta, no entanto, possui uma razoável dificuldade de instalação, e é sugerida apenas como ferramenta secundária.

Quanto ao *Cheops*, que é uma ferramenta específica para localização, o seu uso tende a se tornar mais interessante com o passar do tempo, dado o estado de desenvolvimento. Se no futuro houver a possibilidade de integrar os resultados obtidos nessa ferramenta com as capacidades de gerenciamento do OpenNMS, o administrador de redes terá uma ferramenta completa para gerenciamento de recursos.

Já o *FPING* e o *MTR* são ferramentas muito específicas e limitadas, e podem ser muito bem utilizadas em casos restritos ou como complemento às demais ferramentas.

Capítulo 5

Conclusões

As técnicas e ferramentas apresentadas neste trabalho foram focadas exclusivamente sobre o protocolo TCP/IP versão 4 (também conhecido como IPv4). Entretanto, conforme a Internet vai crescendo uma nova versão desse protocolo vem sedo implantada com o passar dos anos, o IPv6 (RFC 2460 [DEERING/HINDEN]). Além da característica mais notável dessa nova versão que é o número expressivamente maior de endereços IP disponíveis, novas características são adicionadas aos protocolos. No contexto deste trabalho, os mais importantes são o *Neighbor Discovery for IP Version 6 (IPv6)* (RFC 2461 [NARTEN/NORDMARK/SIMPSON]), *Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification* (RFC 3122 [CONTA]) e *Service Location Protocol Modifications for IPv6* (RFC 3111, [GUTTMAN]). Todos são recursos adicionados ao IPv6 que facilitam a localização de elementos na rede, em função da própria característica de mobilidade do protocolo, onde um mesmo endereço IP pode se deslocar entre diversas redes distintas. Com certeza, em um futuro próximo, esses recursos merecem um novo e detalhado trabalho de análise, o que não é possível hoje em função de haverem poucas redes utilizando esse protocolo, e muito menos ferramentas disponíveis.

Outro item citado no decorrer deste trabalho e que merece uma análise mais profunda é o método FCAPS [FUTURESOFT], pois é um conceito extremamente importante e que deveria receber maior divulgação, principalmente por tratar-se da definição de diversas metodologias de gerenciamento de redes. O estudo desses métodos encaixa-se no contexto deste trabalho à medida em que trata de conhecer profundamente a infra-estrutura da rede gerenciada, bem como a validação e verificação contínuas das informações e documentação existentes. As técnicas de *network discovery* podem auxiliar os profissionais seguidores do FCAPS na medida em que provê ferramentas para a verificar principalmente as falhas (localiza-

ção regular ajuda a determinar *hosts* inativos) e questões de segurança (máquinas que não poderiam ser localizadas ou que não deveriam estar na rede, com intrusos ou usários não autorizados). Com certeza o FCAPS merece uma análise à parte e um novo trabalho completamente focado no mesmo.

Muitas outras ferramentas não foram citadas no decorrer deste trabalho, mas merecem menções uma nova pesquisa pessoal por parte do administrador de redes, como as ferramentas de *sniffer* e os analisadores de pacotes e protocolos (principalmente o *tcpdump* e o *ethereal*). Essas ferramentas estão na categoria do *Passive Discovery* e não foram analisadas por ainda não possuir ferramentas acessórias que possibilitem a filtragem dos dados de maneira a delinear as informações desejadas para o *network discovery*. Contudo são ferramentas muito importantes tanto no contexto aqui citado quanto no auxílio para a localização de problemas, falhas de comunicação ou intrusão em redes.

Não se pode também deixar de relembrar sobre as ferramentas Cheops ([SPENCER] e [PRIDDY]), que é muito promissora no ramo do *network discovery* e o OpenNMS ([OPENNMS]), que pode ser uma ferramenta extremamente útil para o gerenciamento de redes.

O *network discovery* pode e deve fazer parte de um processo maior que é a *documentação de rede*. Tal processo, no entando, pode ser bastante trabalhoso e às vezes tedioso. Dessa forma todas as técnicas e ferramentas apresentadas neste trabalho podem auxiliar a, pelo menos, dar início a esse processo. Ele também pode ser utilizado como ferramenta auxiliar no sentido de coletar informações para a *simulação de redes*. A simulação pode auxiliar na previsão de falhas futuras, ou no projeto de expansão ou alterações nas redes.

Infelizmente nenhuma das técnicas pode ser considerada como definitiva, nem mesmo quando aplicadas em conjunto. Todas elas possuem seu valor, importância e lugar dentro da lista de procedimentos e conhecimentos que devem ser adquiridos pelos administradores e pessoal de suporte de rede. No entanto, ainda não existe nenhum *software* que possua inteligência suficiente para resolver todos os problemas e realmente alcançar todos os objetivos ao qual ele foi projetado.

Com esse pensamento o fator humano ainda é extremamente importante, e nesse contexto o conhecimento sobre as técnicas, ferramentas, protocolos envolvidos, vulnerabilidades, sistemas operacionais, etc, precisam todos ser somados e filtrados de forma a criar uma consciência gerencial. É essa consciência que vai determinar a melhor forma de aplicar uma ou mais técnicas e qual ferramenta utilizar, mas que no fim de tudo vai resultar em melhorias na qualidade e no funcionamento da rede gerenciada.

Capítulo 6

Bibliografia

BITWIZARD, B.V. *MTR - A network diagnostic tool*. Novembro 2004. Disponivel em: `http://www.bitwizard.nl/mtr`

BRADEN, R. *Requirements for Internet Hosts - Communication Layers*. Outubro 1989. Disponível em: `http://www.ietf.org/rfc/rfc1122.txt`

BRADNER, S. *The Internet Standards Process – Revision 3*. Outubro 1996. Disponível em `http://www.ietf.org/rfc/rfc2026.txt`

CHAN, H.; CHANG, W.; ESTAN, C. *Projetos ARGUS e OCTOPUS*. Disponível em: `http://www.cs.cornell.edu/cnrg/overview/discovery.html`

CHAPPEL, L. *You're Beign Watched - Cyber-Crime Scans*. Novell Connection Magazine. Março 2001. Disponível em: `http://www.novell.com/connectionmagazine/2001/03/cybercrm31.pdf`

CISCO. *Internetworking Technology Handbook*. Fevereiro 2002. Disponível em: `http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc`

COATES, P. *NOMAD - Network Mapping & Monitoring*. Novembro 2004. Disponivel em: `http://netmon.ncl.ac.uk`

CONTA, A. *Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification*. Junho 2001. Disponível em: `http://www.ietf.org/rfc/rfc3122.txt`

CORBIN, J. *KNetmap - A KDE Network Mapper*. Novembro 2004. Disponivel em: `http://knetmap.sourceforge.net`

CORREIA, L.; SILVA, R. *Redes de Computadores*. Lavras: UFLA/FAEPE, 2002.

CYCLADES. *Guia Internet de Conectividade + Guia de Produtos Cyclades*. São Paulo: Cyclades Brasil, 2001.

DEERING, S.; HINDEN, R. *Internet Protocol, Version 6 (IPv6) Specification*. Dezembro 1998. Disponível em: http://www.ietf.org/rfc/rfc2460.txt

DHAR, S. *Sniffers - Basics and Detection*. Novembro 2004. Disponível em: http://www.rootshell.be/~dhar/downloads/Sniffers.pdf

DZUBIN, T. *FPING - a program to PING hosts in parallel*. Novembro 2004. Disponível em: http://www.fping.com

FARROW, R. *Beware: ICMP can become an attacker's tool for scanning networks*. Maio 2000. Disponível em: http://www.networkmagazine.com/article/NMG20000829S0003

FUTURESOFT. *FCAPS - White Paper*. 2003. Disponível em: http://www.futsoft.com/pdf/fcapswp.pdf

FYODOR. *The Art of Port Scanning*. Setembro 1997. Disponível em: http://www.insecure.org/nmap/nmap_doc.html

GALSTAD, E. *Nagios*. Setembro 2004. Disponível em: http://www.nagios.org

GKANTSIDIS, C. *Experiment and Learn to Discover Network Topology*. Outubro 1999. Disponível em: http://www.cc.gatech.edu/people/home/gantsich/Projects/CS7001_Project1/Report.html

GNU. *GNU General Public License*. Junho 1991. Disponível em: http://www.gnu.org/copyleft/gpl.html

GUTTMAN, E. *Service Location Protocol Modifications for IPv6*. Maio 2001. Disponível em: http://www.ietf.org/rfc/rfc3111.txt

GXSNMP. *Outline of a networked device discovery/configuration engine*. Disponível em: http://www.gxsnmp.org/developers/outline.html

HIPPY, H. *Ping, But No Pong*. Dezembro 2002. Disponível em: http://www.hippy.freeserve.co.uk/nopings.htm

HUNT, C. *TCP/IP Network Administration*. Estados Unidos: O'Reilly & Associates, Dezembro 1997.

HORNING, C. *A Standard for the Transmission of IP Datagrams over Ethernet Networks*. Abril 1984. Disponível em: http://www.ietf.org/rfc/rfc894.txt

HUFFAKER, B.; et al. *Topology discovery by active probing*. 2002. Disponível em: http://citeseer.ist.psu.edu/hu02topology.html

IETF. *Requirements for Internet Hosts - Communication Layers (RFC 1122)*. Outubro 1989. Disponível em: http://www.ietf.org/rfc/rfc1122.txt

INSECURE. *Nmap - Free Security Scanner for Network Exploration & Security Audits*. Novembro 2004. Disponível em: http://insecure.org/nmap

KESSLER, G.; SHEPARD, S. *A Primer On Internet and TCP/IP Tools and Utilities (RFC 2151)*. Junho 1997. Disponível em: `http://www.ietf.org/rfc/rfc2151.txt`

MALKIN, G. *RIP Version 2*. Novembro 1998. Disponível em: `http://www.ietf.org/rfc/rfc2453.txt`

MOCKAPETRIS, P. *Domain Names - Concepts and Facilities*. Novembro 1987. Disponível em: `http://www.ietf.org/rfc/rfc1034.txt`

MOCKAPETRIS, P. *Domain Names - Implementation and Specification*. Novembro 1987. Disponível em: `http://www.ietf.org/rfc/rfc1035.txt`

MOY, J. *OSPF Version 2*. Abril 1998. Disponível em: `http://www.ietf.org/rfc/rfc2328.txt`

NARTEN, T.; NORDMARK, E.; SIMPSON, W. *Neighbor Discovery for IP Version 6 (IPv6) (RFC 2461)*. Dezembro 1998. Disponível em: `http://www.ietf.org/rfc/rfc2461.txt`

NAUGLE, M. *Illustrated TCP/IP*. Estados Unidos: Wiley Computing Publishing, 1998.

OPENNMS. *OpenNMS - Open Network Management System*. Novembro 2004. Disponivel em: `http://www.opennms.org`

OSTG. *Freshmeat*. Novembro 2004. Disponível em: `http://freshmeat.net`

OSTG. *SourceForge*. Novembro 2004. Disponível em: `http://sourceforge.net`

PALMA, L.; PRATES, R. *TCP/IP - Guia de Consulta Rápida*. São Paulo: Novatec Editora, 2000.

PATTERSON, D.; DUAN, Y.; HUANG, D. *Networked System Management: Topology Discovery and Host Monitoring and Control*, Março, 2001. Disponível em: `http://www.cs.berkeley.edu/~duan/prjs/cs252/echo.html`

PLUMMER, D. *An Ethernet Address Resolution Protocol*. Novembro 1982. Disponível em: `http://www.ietf.org/rfc/rfc826.txt`

POSTEL, J. *Internet Control Message Protocol (RFC 792)*. Setembro 1981. Disponível em: `http://www.ietf.org/rfc/rfc792.txt`

PRESUHN, R. *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*. Dezembro de 2002. Disponível em: `http://www.ietf.org/rfc/rfc3416.txt`

PRIDDY, B. *Cheops-ng - the network swiss army knife*. Novembro 2004. Disponível em: `http://cheops-ng.sourceforge.net`

REKHTER, Y.; et al. *A Border Gateway Protocol 4 (BGP-4)*. Março 1995.

Disponível em: http://www.ietf.org/rfc/rfc1771.txt

ROUSE, A. *How to tackle a network documentation project*. Novembro, 2002. Disponível em: http://techrepublic.com.com/5100-6265-1052011.html

SIAMWALLA, R.; SHARMA, R.; KESHAV, S. *Discovering Internet Topology*. Julho, 1998. Disponível em: http://www.cs.cornell.edu/skeshav/papers/discovery.pdf

SIEMENSEN, P. *Nandisc - NLANR Advanced Network Discovery project*. Junho 2001. Disponível em: http://www.scd.ucar.edu/nets/intro/staff/siemsen/nandisc

SPENCER, M. *Cheops Network User Interface*. Novembro 2004. Disponivel em: http://www.marko.net/cheops

STEVENS, W. *TCP/IP Illustrated, Volume 1 - The Protocols*. Estados Unidos: Addison-Wesley, 1994.

VALDUEZA, J.; SIEBES, A. *Scaling Bayesian network discovery through incremental recovery*. Março 1999. Disponível em: http://db.cwi.nl/rapporten/abstract.php?abstractnr=666