Nilmara Goulart Peres Costa

Proposta para migração de um ponto de Rede *Wireless* comercial, de um Provedor de Internet via Rádio, para estrutura configurada em *Software* Livre.

Monografia apresentada ao Curso de Administração em Redes *Linux* (ARL), da Universidade Federal de Lavras, como parte das exigências para obtenção do título de Especialista em Administração em Rede *Linux.*

Orientador
Prof. Msc. Joaquim Quinteiro Uchôa

LAVRAS
MINAS GERAIS - BRASIL
2006

Nilmara Goulart Peres Costa

Proposta para migração de um ponto de Rede *Wireless* comercial, de um Provedor de Internet via Rádio, para estrutura configurada em *Software* Livre.

Monografia apresentada ao Curso de Administração em Redes *Linux* (ARL), da Universidade Federal de Lavras, como parte das exigências para obtenção do título de Especialista em Administração em Rede *Linux.*

Aprovada em 29 de setembro de 2006.

Prof.Msc. Denilson Vedoveto Martins

Prof.Msc. Simone Markenson Pech

Prof.Msc. Joaquim Quinteiro Uchôa
(Orientador)

Lavras
Minas Gerais - Brasil

Agradecimentos

Agradeço, primeiramente, a Deus e ao Divino Espírito Santo que me iluminaram muitas vezes mostrando o caminho das pedras; especialmente ao meu esposo Durval e aos meus filhos Vitor e Vinícius pelos sábados, domingos e feriados...ao Patrick Brandão que me ajudou com dicas, ao Sérgio Antônio que muito contribuiu e ao o Prof. Msc. Joaquim pela orientação e atenção prestada.

Sumário

Lista de Figuras

Resumo

O objetivo desta monografia é apresentar uma proposta sobre a migração de uma estrutura de Rede *Wireless* de modelo comercial, de um Provedor de Internet via Rádio, para uma estrutura baseada e configurada em *Linux*. Esta estrutura de rede que chamaremos de APlinux, tem como propósito, conectar clientes de um Provedor Banda Larga via Rádio à internet, com autenticação dos referidos usuários. O projeto visa baixar gastos com aquisição de equipamentos, reduzir custos com licenças de utilização de *Softwares*, independência em relação à assistência técnica e apresentar um sistema que suporte um número maior de usuários conectados simultaneamente. Este trabalho abrangerá a instalação do *Linux* para configuração do APLinux, configuração de *Firewall*, autenticação de usuários e controle de banda, utilizando um exemplo prático e real, que mostrará os benefícios alcançados com a adoção de *Software* Livre.

Capítulo 1

Introdução

As *WLANs (Wireless Local Area Network)* - um dos vários tipos de redes locais sem fio, proporcionam mobilidade e conexões em altas velocidades, que combinadas com preços mais acessíveis, estão tornando-se populares. A expansão da utilização deste tipo de rede, pode ser notado também, pelo fato de que vários modelos de *notebook* já virem equipados com placas de redes *wireless*[1].

O funcionamento das *WLANs* pode ser através da utilização de equipamentos denominados rádios ou *Access Point*. Através dos rádios são estabelecidas as comunicações de dados entre pontos da rede. Os dados são modulados no rádio e transmitidos através de ondas eletromagnéticas. Para mais detalhes sobre ondas eletromagnéticas, consulte Sanches (2005). A figura 1.1 mostra um dos vários modelos de funcionamento de uma *WLAN* via rádio.

Figura 1.1: Mapa de uma rede *Wireless*

[1] *Wireless* - Conexão sem fio, que utiliza o ar como meio de tranporte de informações.

As *WLANs* podem atender desde conexões ponto a ponto como, por exemplo, interligar uma empresa matriz com sua filial, ou mesmo, satisfazer às finalidades de uma rede integrada com várias filiais ou empresas que necessitam comunicar-se entre si em alta velocidade. Existem vários provedores que adotam essa tecnologia para prover acesso à internet aos seus usuários, tais como: empresas, escolas, condomínios, usuários domésticos, etc.

Para operar legalmente, o Provedor de Internet via Rádio, precisa de uma licença de SCM[2], emitida pela ANATEL (Agência Nacional de Telecomunicações).

Os benefícios das redes *WLANs* e suas facilidades de implantação, são fatores que a tornam vulneráveis, segundo Rufino (2005). Os sinais são transmitidos pelo ar, proporcionado alcance dos mesmos através de antenas e os ataques podem ser feitos com mais facilidade e com menos conhecimento. Os perigos e riscos de transmitirem dados, via rede *WLAN,* são, dentre outros: espionagem dos dados, interceptação e modificação de dados transmitidos, acesso imediato à rede por um usuário mal intencionado.

Neste contexto, o presente trabalho pretende apresentar uma proposta de migração de um ponto de Rede *Wireless* de modelo comercial, para uma estrutura configurada em *Linux.* Isso inclui autenticação de usuários, controle de banda, controle de *IP* e *MAC*[3], para o Provedor Netcoro de Internet via Rádio, situado em Coromandel - MG, na região do Alto Paranaíba.

A presente monografia encontra-se estruturada da seguinte forma: Capítulo 2 apresenta a estrutura da rede física do Provedor Netcoro, de acesso à internet, incluindo problemas encontrados e enfrentados na atual estrutura, definindo itens motivadores para este trabalho; o Capítulo 3 aborda sobre as ferramentas escolhidas e suas respectivas

---

[2] Serviço de Comunicação Multimídia é um serviço fixo de telecomunicações de interesse coletivo, prestado em âmbito nacional e internacional, no regime privado, que possibilita a oferta de capacidade de transmissão, emissão e recepção de informações multimídia, utilizando quaisquer meios, a assinantes dentro de uma área de prestação de serviço. Informações sobre assunto poderão ser encontradas em *http://www.anatel.gov.br*

[3] *MAC Adress*: Endereço Físico de um componente de rede

funções, para o desenvolvimento do trabalho; o Capítulo 4 trata-se da instalação do *Linux*, da configuração do cartão *Wireless* em modo *AP*[4], da configuração e instalação de programas para controle de usuários e da segurança da rede sem fio; o Capítulo 5 inclui a implantação da nova estrutura de Rede *Wireless* e por fim, o Capítulo 6 apresenta as conclusões deste trabalho.

[4] *Access Point(AP)* - Ponto de acesso. O Ponto de Acesso tem função semelhande à função que o *hub* desempenha nas redes com fios: retransmitir os pacotes de dados, de forma que todos os computadores da rede os recebam.

Capítulo 2

Histórico, contextualização e problematização

2 Netcoro - Provedor de Acesso a Internet via rádio

2.1 Histórico

O Provedor Netcoro, incialmente com o nome de CerradoNet, iniciou suas atividades no ano de 1997, na cidade de Coromandel - MG..

Sua estrutura inicial era a seguinte: Servidor *Windows NT* ligado a um roteador que recebia um *link* de velocidade de 64k de um provedor de uma cidade vizinha. Nesta época, o acesso dos usuários era via linha discada, ou seja, *Dial Up*. A conexão aos usuários era feita através de uma Multiserial interligada a modens *US Robotics*. Em 1998, o servidor foi migrado para plataforma *Linux* devido a constantes travamentos do *Windows NT.* Neste período, o *link* passou a ser local, atendido diretamente pela concessonária. Em 1999, a Multiserial foi substituída por um *Patch Router*. Em 2000, foram instaladas linhas digitais e o acesso era via PR 4000. Em 2003, iniciou-se o atendimento via Rádio, conectando os primeiros usuários. A partir daí, a rede via Rádio foi se expandindo em outros pontos e vários equipamentos, de modelo comercial, foram sendo adquiridos. A figura 2.1 mostra a atual estrutura de um dos pontos de acessos da rede .

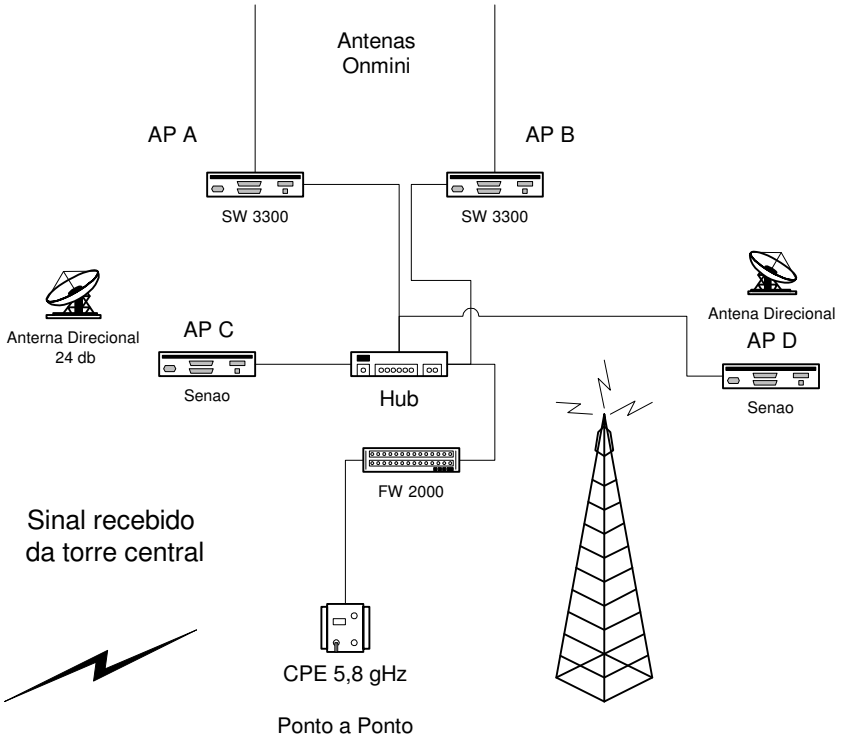

Figura 2.1: Ponto de Acesso do Provedor Netcoro

2.2 Problemas enfrentados na atual estrutura

O equipamento FW 2000, de modelo comercial, funciona como *Firewall, DHCP* e controle de banda. No último período de chuvas, foram danificados dois equipamentos deste modelo, por causa de sobretensão na rede da concessionária de energia elétrica, causando interrupção no atendimento aos usuários. Naquele momento, os equipamentos danificados foram substituídos por uma máquina com o sistema *StarOs*, que estava em fase de testes e que cumpriu satisfatoriamente o seu papel. O *StarOs* ficou em funcionamento até o retorno dos FW 2000.

O equipamento FW 2000 é comercializado por uma única empresa no Brasil, situada em São Paulo, capital, dificultando reposição ou manutenção rápida. Além disto, a negociação para aquisição destes equipamentos torna-se difícil, por não conhecermos outra empresa que ofereça o mesmo produto. Outro problema, detectado recentemente, foi o surgimento de usuário não autorizado na rede, com *MAC* clonado. Esse problema foi comprovado com testes feito em laboratório. Essa clonagem é bastante simples de ser realizada, e além disso, o sistema atual não consegue identificar automaticamente essa falha de segurança, permitindo que pessoas não autorizadas naveguem livremente na rede, sem serem percebidas.

Tendo em vista os problemas apresentados, a proposta do presente trabalho é a configuração de um servidor *Linux*, que possibilite, além das funcionalidades apresentadas no sistema atual, a implantação de um sistema de gerenciamento de usuários. Este sistema permitirá a autenticação, autorização e contabilização de acesso de usuários na rede, obtendo, por exemplo, informações sobre a origem das conexões, tentativas de acessos bem ou mal sucedidas, o horário que o usuário conectou e o período que permaneceu na rede.

O novo sistema, visa ainda, baixar custo com aquisição de equipamentos e a obtenção de um sistema de *backup*, para eventual substituição de equipamentos em situações emergenciais. Obtém-se com isso, uma maior independência do provedor, que poderá agir com mais rapidez na solução de problemas que surgem na rede. Outras vantagens, a serem apresentadas no novo sistema, referem-se na economia em relação a assitência técnica e reposição de peças e equipamentos, pelo fato do sistema ser instalado em computadores, o que facilita a manutenção do mesmo.

Capítulo 3

Ferramentas e protocolos utilizados

Na escolha das ferramentas, para o desenvolvimento do projeto, foram considerados os itens de confiabilidade, robustez e gratuidade dos programas.

3.1 Sistema Operacional

O sistema operacional adotado, foi o *Linux* distribuição *Slackware* pelo fato de que, os servidores que se encontram no Provedor, já estarem configurados nesta distribuição e terem sua eficiência comprovada, ao longo de seis anos de utilização. Além disso, trata-se de uma distribuição robusta e enxuta.

3.2 Bandlimit

Para controlar o uso da largura de banda dos usuários, foi utilizado o *script Bandlimit*[1]. Este *script* foi escolhido, por se tratar de uma implementação funcional, gratuita e de fácil configuração. O *Bandlimit* usa *CBQ*[2] e *Iptables*[3] para marcar e limitar banda de IP válido e inválido na rede.

Uma outra opção a ser considerada, futuramente, será o *HTB*[4], por se tratar de uma ferramenta precisa e que possui mais recursos como, por exemplo, quando uma classe requisita menos banda do que o total a ela reservada, a largura de banda remanescente é distribuída para outras classes que requisitaram serviços.

---

[1] *Bandlimint* - É um script que tem a função de limitar banda de *donwload* e *upload*. O *Bandlimit* pode ser encontrado em *http://wiki.underlinux.com.br/index.php/Projetos/BandLimit*.

[2] *CBQ - Class Based Queue - Script* para controle de banda. Pode se adquirido através do site *https://sourceforge.net/projects/cbqinit* .

[3] *Iptables* - Informações sobre o *Iptables* podem ser acessadas através do endereço *http:// www.netfilter.org*

[4] *HTB (Hierarchical Token Bucket)* - Informações sobre o *HTB* poderão ser acessadas em *http:/ /luxik.cdi.cz/~devik/qos/htb/.*

3.3 Iptables

O *Iptables* atua como uma ferramenta de segurança, trabalhando com filtragem de pacotes e é utilizado para criar as regras de *Firewall e Nat.* O *Iptables pod*e ser configurado para filtrar e até alterar dados empacotados, com base em diversos critérios, como endereço de origem e destino de pacotes. Com o *Iptables* pode-se implementar recursos no *Kernel*, como a ispenção de posição de pacotes, aumentando muito a habilidade de um *firewall* e consequentemente, protegendo o computador onde o mesmo foi instalado e configurado.

3.4 Chillispot

Para controle de acesso de usuários na rede algumas ferramentas foram pesquisadas, como: *NoCAT* (Valiñas 2006), *Oasis* (Oasis 2002) e o *Chillispot*[5]. O *Chillispot* foi escolhido pelo fato do mesmo apresentar algumas funcionalidades adicionais e ser um projeto mais atualizado do que os outros.

O *Chillispot* é um programa livre de controle de acesso a redes sem fio e foi utilizado para autenticação de usuários, através de um navegador *web*, com o propósito de evitar ou amenizar o acesso a rede de usuários não autorizados.

O *ChilliSpot* trabalha com o *DHCP*[6] e este pode ser utilizado para fornecer, automaticamente, as configurações da rede para os usuários, de forma a facilitar a administração e o suporte do provedor.

3.5 Radius

O Radius é um protocolo amplamente empregado para disponibilizar acesso a redes com autenticação, autorização e contabilização. Ele foi desenvolvido, originalmente

[5] *ChilliSpot* - Informações adicionais podem serem encontradas em *http://www.chillispot.org.*
[6] *DHCP - Dynamic Host Configuration Protocol* - protocolo dinâmico de configuração de *host*

para uso em serviços de acesso discado, mas, pela sua simplicidade, eficiência e facilidade de implementação, hoje é suportado por servidores de *VPN*, *AP's* e outros tipos de acesso a redes.

O Radius utilizado neste trabalho, para gerenciar as tarefas de autenticação e contabilização, foi o *FreeRadius*[7].

3.6 *PHP* , *MySQL* e *phpMyAdmin*

"PHP é acrônimo de *Hypertext Prerpocessador* (pré-processador de hipertexto), uma poderosa linguagem de programação *open source*, mundialmente utilizada, principalmente no ambiente *web*..." (Soares, 2004).

Com a utilização dos programas *phpRADmin* e *phpMyAdmin* foi necessário a instalação do programa *PHP*, para desenvolvimento do trabalho.

Para armazenamendo de dados dos usuários e trabalhar em conjunto com o *FreeRadius* e *phpRADmin*, foi adotado o *MySQL*[8] que é um *SGBD* de código aberto. O *MySQL* trabalha com relacionamento entre tabelas e atende bem às demandas de sistemas de produção de média escala. Segundo Soares(2004), dentre os vários bancos de dados suportados pelo PHP, o *MySQL* é o mais utilizado.

O *phpMyAdmin*[9] foi utilizado para facilitar o gerenciamento das tabelas dos bancos de dados utilizados no sistema. O *phpMyAdmin* possui uma *interface web* que permite criar, consultar e apagar tabelas de um banco de dados. O *phpMyAdmin* executa instruções *SQL*, gerencia chaves do campos das tabelas, permite importar e exportar dados, etc.

---

[7] *FreeRADIUS* - Programa utilizado para autenticação e contabilização de acesso de clientes à rede. O *FreeRadius* pode ser encontrado em *http://www.freeradius.org/* .

[8] *MySQL* - é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Structured Query Language - Linguagem de Consulta Estruturada) como interface. O *MySQL* pode ser encontrado em *http://www.mysql.org/.*

[9] *phpMyAdmin* - Disponível em *http://www.phpmyadmin.net.*

3.7 Apache

Como o *ChilliSpot* autentica os clientes via *browser*, o *Apache*[10] foi escolhido para ser o servidor *Web* do sistema. Segundo Smith(2003) como regra geral, *Apache* é uma boa escolha para um servidor *Web*, devido à sua popularidade e ao fato de que ele vem com todas as principais distribuições *Linux*. O motivo desta escolha é que o *Apache* possui vários recursos e suporta o conjunto de opções necessárias para a implementação do *ChilliSpot* como scripts *GGI* e segurança *SSL*.

3.8 Monitoração e administração

Para facilitar a administração dos usuários, foi utilizado o *phpRADmin* um programa gerado em *PHP,* utilizado para administração de um servidor Radius. Ele possui controle sobre as tabelas do Radius, para alterar atributos, administrar usuários, grupos e uma interface básica de administração de cadastro de clientes.

Com o *phpRADmin*[11], tem-se a criação de gráficos de monitoramento a partir de informações armazenadas no banco de dados do Radius. Estes gráficos, podem-se ter controle sobre quantas sessões são abertas em horários diferentes, controle sobre o número de registros no banco de dados, monitoração da quantidade de banda consumida pelos usuários, etc.

---

[10] *Apache* - Servidor Web normalmente é o pacote instalado por padrão quando se solicita a instalação de um servidor *Web.* O principal Web site do *Apache* é *http://httpd.apache.org.*

[11] *phpRADmin*- É uma ferramenta escrita em *PHP*, para administrar o servidor *FreeRADIUS* via *Web*. Esta interface permite o administrador configurar, buscar, criar e editar usuários em uma base de dados. Sua *interface* incorpora gráficos com informações de usuários facilitando o gerenciamento da rede. O *phpRADmin* está preparado para funcionar com o *Chillispot*. O programa pode ser encontrado em http://www.phpradmin.org/

Capítulo 4

Instalação e configuração das ferramentas

4 Instalação e configuração

Neste capítulo, apresentaremos a parte operacional do projeto, ou seja, a forma que foi usada para a instalação e configuração das ferramentas citadas no Capítulo 3.

4.1 Instalação do Sistema Operacional

A distribuição *Linux* escolhida foi a *Slackware*[1] versão 10.2, como foi justificado no item 3.1. O *Slackware* é o nome de uma das mais antigas e conhecidas distribuições do *Linux*. Seu objetivo é manter-se fiel aos padrões *UNIX* e além disso, ela é composta de aplicativos estáveis.

No *Slackware* foi utilizado o *Kernel* 2.4.32, instalado em uma máquina Intel Pentium II com 128MB RAM, HD 10 GB.

4.2 *Iptables, Patch-o-Matic e Layer 7*

O programa utilizado para configurar o *firewall* foi o *Iptables,* que pode ser acessado em http://www.netfilter.org/projects/iptables/files/iptables-1.3.5.tar.bz2.

O *Patch-o-Matic* foi utilizado para aplicar *pacth*[2] ao *Iptables*. O *Pach-o-Matic* pode ser encontrado em *http://ftp.netfilter.org/pub/patch-o-matic-ng/snapshot/patch-o-matic-ng-20050918.tar.bz2.*

As funcionalidades aplicadas com o *Patch-o-Matic* foram: CLASSIF,

---

[1]*Slackware* - Suas versões poderão ser encontradas em *ftp://ftp.slackware-brasil.com.br/* e instalações e configurações passo a passo , podem ser acessadas em *ftp://ftp.slackbook.org/pub/slackbook/slackbook-2.0.pdf.* Site oficial http://www.slackware.com/.

[2]*Patch* - São funcionalidades ainda não incorporadas aos fontes originias do *Iptables.*

CONNMARK, time, TTL, unclean, expire, iprange, psd, quota, MARK e IPMARK. Para maiores destalhes consulte (Souza, 2003).

O *Layer 7*[3] é um filtro de pacotes que se baseia no protocolo da aplicação utilizada. Com aplicação do Layer 7, o *Iptables,* passa a suportar mais funcionalidades, como por exemplo, bloquear os aplicativos *MSN, Yahoo Messenger, ICQ*, filtrar pacotes do *Kazaa, HTTP, Jabber, Citrix, Bittorrent, FTP, Gnucleus*, etc.

4.3 Compilação do *Kernel*

Após a instalação e aplicação das funcionalidades com o *Patch-o-Matic,* necessárias para o andamento do trabalho, foi preciso fazer a compilação do *Kernel*[4]. Para isso, foram ativados alguns módulos, como mostra a figura 4.1.

```
# cd /usr/src/linux

# make menuconfig

# ----> NETWORKING OPTIONS --> Marcar todas as opções em branco com M
(de módulo) com execeção de Layer 7 debugging output (experimental) ipfwadm (2.0-
style) support . Por se tratar de módulo desnecessário aos nossos objetivos.

# make dep

# make bzImage

# cp archi386/boot/bzImage /boot/linux-2.4.32.NF
```

Figura 4.1: Ativação de módulo e compilação do *Kernel*

[3] *Layer 7* - Pode ser encontrado em *http://ufpr.dl.souceforge.net/sourceforge/l7-filter/netfilter-layer7-v2.1.tar.gz* e os protocolos do *Layer 7* podem ser acessados em em *http://ufpr.dl.sourceforge.net/sourceforge/l7-filter/l7-protocols-2006-03-13.tar.gz*

[4] *Kernel* de um sistema operacional é entendido como o núcleo deste ou, numa tradução literal, *cerne*. Ele representa a camada mais baixa de *interface* com o *Hardware*, sendo responsável por gerenciar os recursos do sistema computacional como um todo. É no *kernel* que estão definidas funções para operação com periféricos (*mouse*, discos, impressoras, *interface* serial/*interface* paralela), gerenciamento de memória, entre outros. Resumidamente, o *kernel* é um conjunto de programas que fornece para os programas de usuário (aplicativos) uma *interface* para a utilização dos recursos do sistema.

Ao término da compilação do *Kernel,* foram realizadas as aplicação de *path* do Layer 7.

/usr/src/iptables# patch -p1 < /usr/layer7/itables-layer7-2.1.patch

#chmod +x extensions/.layer7-test

#chmod +x extensions/.expire-test

#chmod +x extensions/.expire-test6

#make KERNEL_DIR=/usr/scrc/linux

#make install KERNEL_DIR=/usr/src/linux

4.4 Compilação e instalação do IPP2P

A compilação e instalação o IPP2P seguiram conforme os passos:

cd /usr/src/linux/ipp2p

Devido a mudança do nome do diretório do *Iptables,* houve necessidade de alteração de uma linha do arquivo *Makefile*, conforme a seguir:

IPTABLES_SRC=/usr/src/iptables-1.2.9

para:

IPTABLES_SRC=/usr/src/iptables

make

cp libipt_ipp2p.so /usr/local/lib/iptables

Para acertar os módulos, foram acrescentadas algumas linhas dentro do arquivo */etc/rc.d/rc.molules*.

/sbin/modprobe ip_tables

/sbin/insmod usr/src/ipp2p/ipt_ipp2p.o

/sbin/modprobe ipt_layer7

A compilação e instalação dos módulos foram realizados conforme a seguir:

/usr/src/linux# mv /lib/modules/2.4.32 /lib/modules/2.4.32-old

#make modules

make modules_install

4.5 Configuração do cartão *Wireless* em modo *master*

Em substituição ao *Access Point,* de modelo comercial, foi configurado um cartão *Wireless,* marca *Senao,* para trabalhar em modo *master*[5].

O driver para este cartão pode ser encontrado em *http://hostap.epitest.fi/releases/ hostap.driver-0.4.9.tar.gz.*

Instalação realizada para que o cartão *wireless* opere em modo *master*:

/etc/rc.d/rc.pcmcia stop

tar -zxvf hostap-driver-0.4.7.tar.gz

make

#make install

#modprobe hostap_pci

4.6 Implementação da Chave WEP

O *WEP - Wired Equivalent Privacy* foi o primeiro recurso de segurança disponibilizado para redes *wireless*. "O *WEP* possui algumas vulnerabilidades. Alguns programas, já largamente disponíveis, são capazes de quebrar as chaves de codificação caso seja possível monitorar o tráfego da rede durante algumas horas."(Sanches, 2005).

[5] Modo *Master* - Nesse modo de operação, o cartão *wireless* passa a operar como um ponto de acesso.

Criado para corrigir a segurança do sistema *WEP,* o *WAP,* segundo Sanches (2005), implementa a maior parte das exigências do padrão de segurança e foi criado como uma medida intermediária para ocupar o lugar do *WEP*.

Apesar de vários artigos e livros descreverem sobre a fragilidade da chave *WEP,* a mesma foi implementada no sistema para auxiliar na segurança da rede.

O cartão *SENAO*, utilizado no trabalho, não suportou a chave *WAP*, que é uma chave mais segura do que a *WEP*. A configuração da chave *WEP* foi realizada no arquivo /etc/rc.d/inet1.conf e a seguinte linha foi configurada:.

WLAN_KEY[4]=Aqui foi digitada a chave WEP de 128 bits contendo digitos alfanuméricos.

Estudos para aplicação de chave *WAP* em outro modelo de cartão *Wireless,* estão em andamento, apesar de serem também vulneráveis. Por outro lado, as chaves mencionadas, mesmo consideradas sem segurança, se forem usadas em conjunto com outras ferramentas, dificultam o acesso não permitido à rede *Wireless*.

4.7 Controle de Banda

Para controlar a banda utilizada pelos usuários, foi instalado o *script Bandlimit.* A seguir são exibidos os passos para instalação do *Bandlimit*.

Foi criado o diretório *bandlimit* dentro do /etc

mkdir /etc/bandlimit

Dentro do diretório *bandlimit* foram criados os aquivos *ips* e *interfaces.*

touch /etc/bandlimit/ips

touch /etc/bandlimit/interfaces

Os arquivos *ips* e o *interfaces* foram editados com os respectivos números de *IP* e nomes das *interfaces* utilizadas no sistema.

Os IPs a serem limitados foram configurados no arquivo *ips* no formato abaixo:

ip:ratein:rateout ex: 192.168.170.2:64:64, ou seja, número do IP e tamanho da banda de entrada e saída.

Foi configurado, dentro do arquivo *interfaces,* as *interfaces* utilizadas no sistema.

no formato ethx ex: eth0

Para funcionar automaticamente o *script*, o comando de execução do *Bandlimit* foi adicionado ao arquivo /etc/rc.d/rc.local

rc.bandlimit start

4.8 Instalação do ChilliSpot

O programa *ChilliSpot*[1] tem por finalidade evitar que pessoas não cadastradas ou não autorizadas, tenham acesso a Internet via rede *Wireless*. Ele captura o tráfego na porta 80 e o desvia para um servidor HTTP. Neste servidor, uma página de autenticação de usuários, solicitará uma senha e nome de usuário. A autenticação dos usuários no *Chillispot* é feita através de um servidor Radius que gerencia as tarefas de autenticação e contabilização. Após a autenticação bem sucedida, o acesso à Internet será liberado através de regras do *firewall*.

O *ChilliSpot* suporta os Radius *FreeRADIUS*[2] , *OpenRadius, Cistron* e *IC-Radius* que são *Open Source,* dentre outros. Para autenticação de usuários foi utilizado o *FreeRADIUS.*

[1] Informações sobre o *ChilliSpot* pode ser encontrado em http://www.chillispot.org.

[2]Maiores informações sobre o *FreeRADIUS* poderão ser acessadas no site http:// www.freeradius.org.

O *ChilliSpot* é composto basicamente por dois módulos: *Hostsplotlogon.cgi* , que é um formulário web de autenticação de usuários e o *Chilli* que é o *daemon*[3] do sistema. O *Chilli* possui um arquivo básico de configuração onde ficam definidos a rede a ser utilizada, configurações do rádio, configurações do *DHCP*, etc. O *Chilli* trabalha usando driver TUN/TAP como *interface* virtual da rede.

Na página de autenticação do usuário é utilizado o protocolo *HTTPS*. Por este motivo, a instalação do *Apache* é necessária para a utilização da página de autenticação, utilizando de suas caracteríticas de segurança implementadas como *SSL (Secure Socket Layer),* que garante um túnel seguro de comunicação e a troca mútua de certificados digitais.

4.8.1 Instalação e arquivos de configuração do ChilliSpot

A instalação do ChilliSpot foi feita da seguinte forma:

cd /usr/local/src

wget http://www.chillispot.org/download/chillispot-1.0.tar.gz

tar zxvf chillispot-1.0.tar.gz

O *Chillispot* possui um arquivo de configuração que deverá ser editado com as opções necessárias para o funcionamento da rede *Wireless*.

A seguir são exibidas as alterações realizadas no arquivo /etc/chilli.conf.

net 192.168.190.0/255.255.255.0

dns1 200.216.210.34

dns2 200.216.210.35

domain compucenternet.com.br

---

[3] Daemon - É uma aplicação que é executada no sistema.

radiuslisten 127.0.0.1

radiusserver1 127.0.0.1

radiusserver2 127.0.0.1

radiusauthport 1812

radiussecret segredoradius

dhcpif wlan0

uamserver https://192.168.170.1/cgi-bin/hotspotlogin.cgi

uamhomepage https://192.168.170.1/welcome.html

uamsecret segredochilli

Na opção *uamsecret,* a senha segredochilli, como mostra nosso exemplo, deverá ser idêntica à senha que se encontra no arquivo *hotspotlogin.cgi* na opção $uamsecret = "segredochilli".

A mesma forma se aplica para a opção *radiussecret,* mas, no entanto, a senha do Radius deve ser localizada no arquivo de configuração do Radius utilizado, no caso do *FreeRADIUS,* fica no arquivo *clients.conf*

4.9 Instalação do Apache e criação de certificado

A seguir, são listados os comandos utilizados para a instalação do *Apache*, com ativação das opções necessárias como *CGI* e *SSL*.

```
# cd /usr/local/src
# wget http://apache.usp.br/httpd/httpd-2.0.50.tar.gz
# tar xvzf httpd-2.0.50.tar.gz
# cd httpd-2.0.50
# ./configure —enable-ssl —enable-cgi —enable-suexec —enable-so
# make
# make install
```

4.9.1 Criação de chaves e de certificado

Para criação de chaves e certificado, foram utilizados os seguintes comandos:

```
# cd /usr/local/apache2
# mkdir ssl
# cd ssl
```

4.9.1.1 Criando o certificado

```
# openssl genrsa -out mono.key 1024
```

4.9.1.2 Criando assinatura do certificado digital

```
# openssl req -new -key nocat.key -out mono.csr
```

4.9.1.3 Certificado auto-assinado

```
# openssl x509 -days 365 -req -in  mono.csr -signkey nocat.key -out mono.crt
```

4.10 Instalação do Freeradius

A instalação do *FreeRAdius* foi realizada através dos comandos:

#./configure

make

make install

O *FreeRADIUS* foi utilizado em conjunto com o *Mysql* para autenticação dos usuários. Foram implementadas as opções que não permitem conexões simultâneas de clientes, atributos para que o usuário tenha acesso à internet utilizando somente o MAC e IP cadastrados para a máquina do referido cliente.

4.10.1 Criação das tabelas do FreeRADIUS

O *FreeRADIUS* possui a DDL[4] pronta, que normalmente se encontra armazenada em /usr/local/src/freeradius-1.0.1/src/modules/rlm_sql/drivers/rlm_sql_mysql.

mysqladmim -psenharoot create radius

mysql -psenahroot radius < db_mysql.sql

Exemplo de cadastro no banco de dados radius, para negar conexão simultânea de usuários:

INSERT into usergroup (username, groupname) VALUES (`joao`,`sessaounica`);

4.11 Instalação do *PhpRADmin*

Para instalação do *PhpRADmin* foram adicionadas algumas configuração no arquivo /etc/apache/httpd.conf, pois a instalação e configuração do *PhpRADmin* é feita via *browser* por meio do arquivo /var/www/phpradmin/www/install/setup.php

Através do arquivo setup.php, deverão ser seguidos, basicamente, oito passos bastante intuitivos, onde deve-se informar os dados necessários para o funcinamento do programa, tais como: aceitação da licença, configuracões de diretórios, verificação de status, configurações de senhas de acesso ao banco de dados e ao sistema, etc.

Logo após a instalação, deve-se fazer alguns ajustes no arquivo /usr/local/phpradmin/conf/dialup_admin/conf/admin.conf, além de algumas inserções de configurações no arquivo crontab e alteração de permissão de acesso em alguns arquivos. Essas modificações encontram-se relacionadas no útimo passo da instalação.

[4] DDL (Data Definition Language) são rotinas contendo instruções para criação de tabelas, campos, etc, de um banco de dados.

Capítulo 5

Testes, implantação e gerenciamento

5.1 Estrutura da nova rede

Após a instalação e configuração do sistema, o mesmo foi colocado em fase de testes. A figura 5.1 simula um dos pontos da nova estrutura da redel.

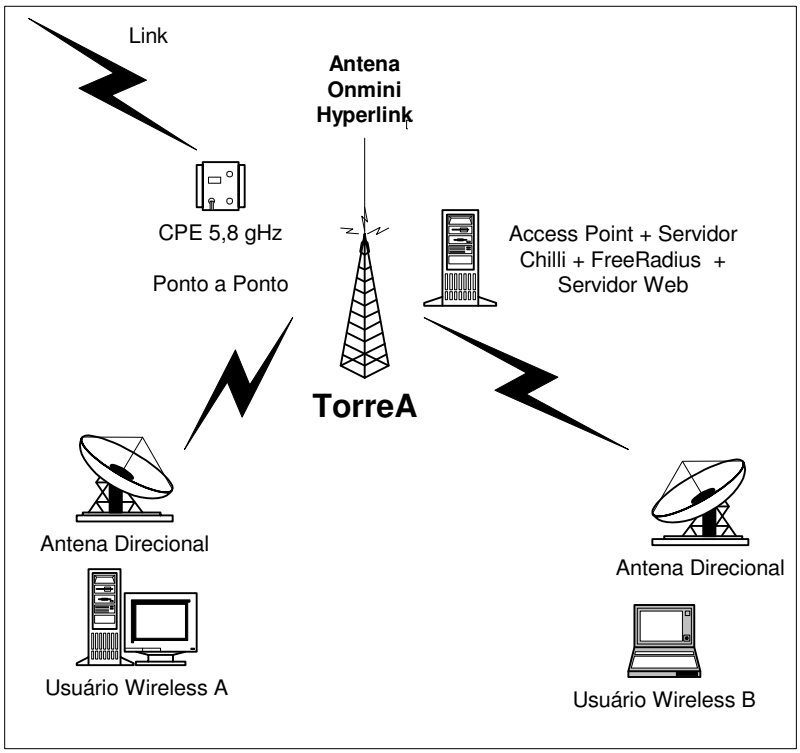

Figura 5.1: Mapa da rede com a estrutura *APLinux*

5.2 Testes e implantação

Os primeiros testes foram realizados em um computador de *desktop*, com uma placa *pci wireless* e um *notebook* com cartão *wireless* interno, ambos com *Windows XP*, pelo fato que a maioria dos clientes possuem o referido programa instalado em suas

máquinas.

A figura 5.2 exibe a tela de conexão de sinal do APlinux com o usuário. Como a chave *WEP* foi configurada no sistema, um desenho de um cadeado é apresentado como mostra a figura 5.2.

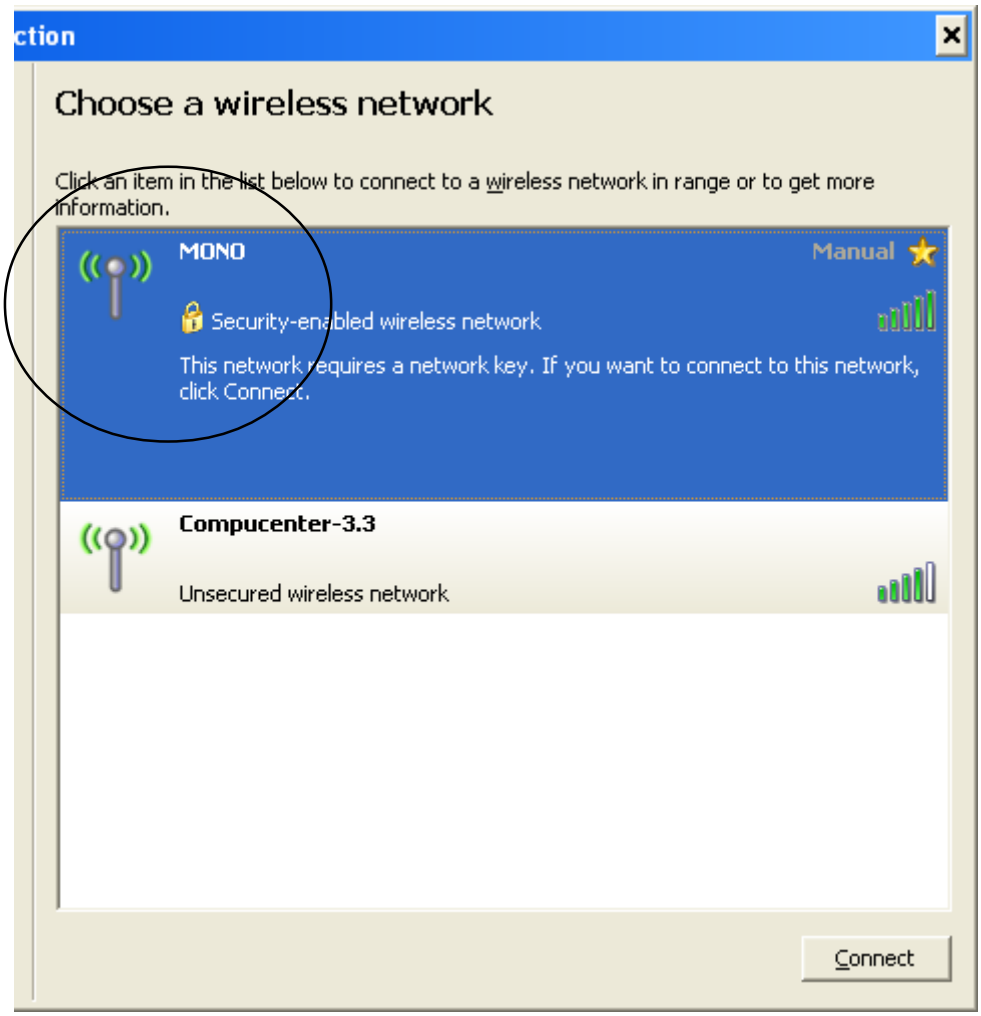

Figura 5.2: Tela exibindo a APlinux com utilização de chave *WEP*

Ao tentar conectar-se no APlinux pela primeira vez, será exibida ao usuário, uma tela solicitando a chave *WEP* do APlinux , como ilustra a figura 5.3. Desta forma, após a digitação e confirmação da chave, a mesma ficará gravada automaticamente.

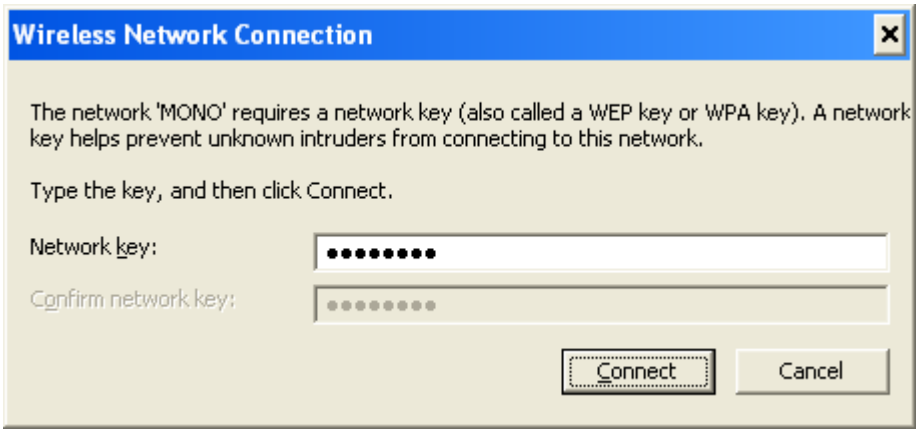

Figura 5.3: Tela exibindo solicitação de senha da chave *WEP*

A próxima figura, exibe o usuário conectado no APlinux, porém nenhum serviço de rede foi liberado, como pode ser constatado nos ícones do *Skype* e *MSN*, na barra apresentada na figura 5.4.

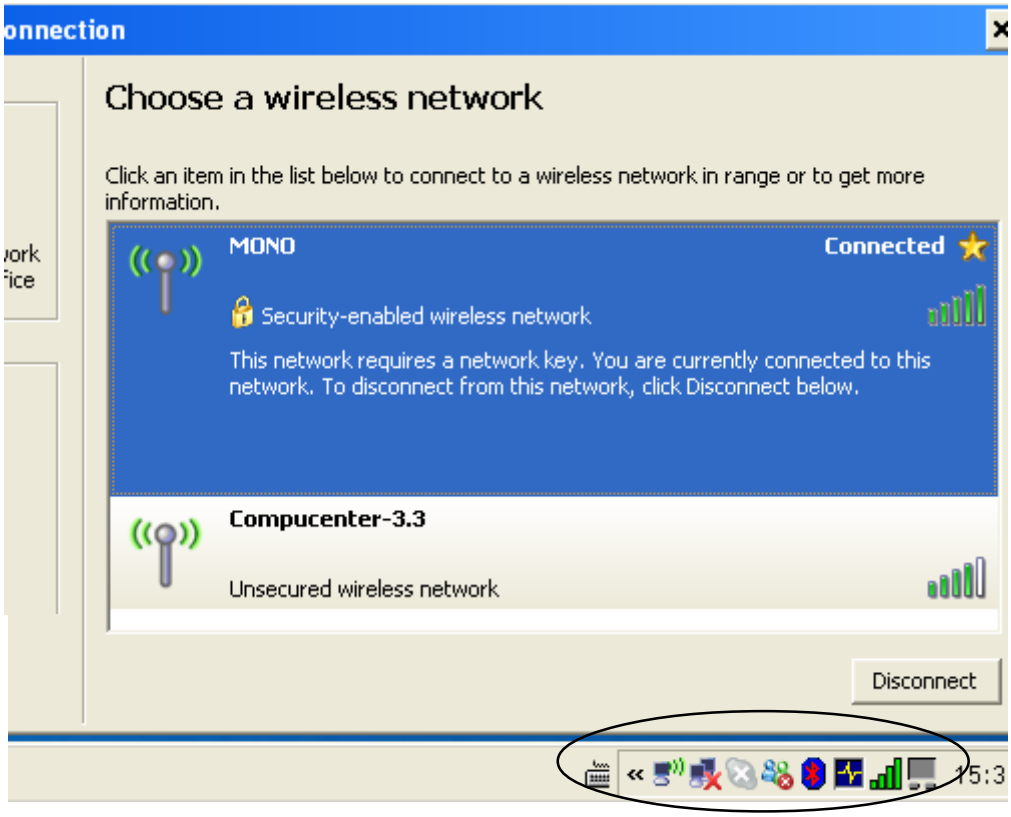

Figura 5.4: Cliente conectado no APlinux sem liberação de serviços

A figura 5.5 mostra a tela de conexão que será exibida ao usuário, para fazer autenticação no sistema.

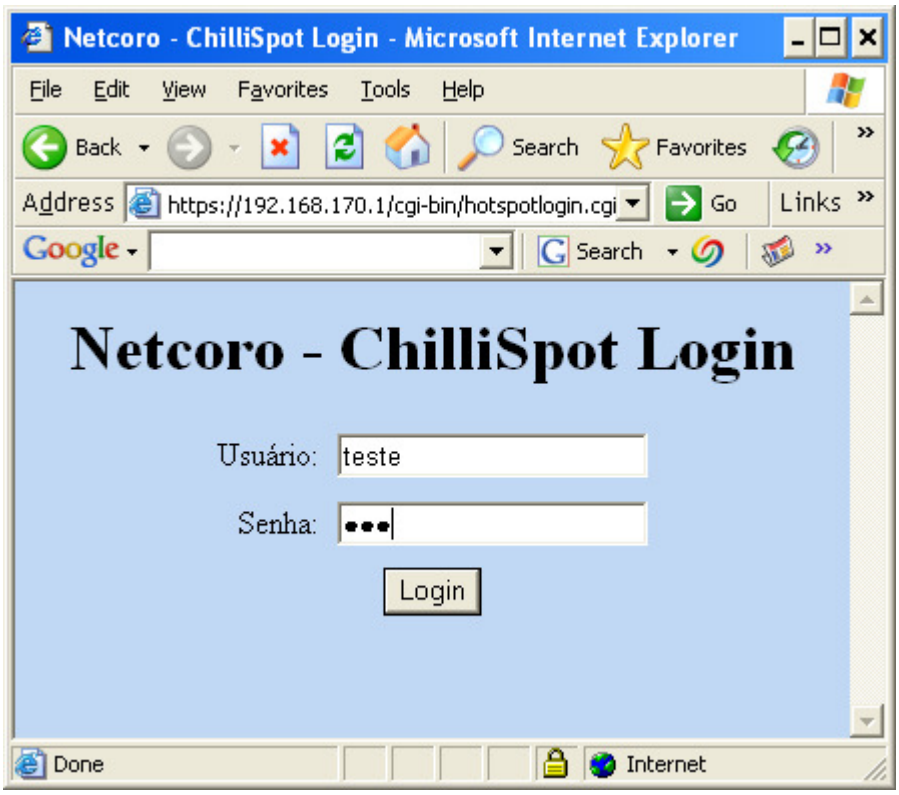

Figura 5.5: Página de autenticação de usuários

Para que os serviços e portas de acesso à Internet sejam liberados, o usuário deverá acessar o *browser* e através de uma página inicial configurada no *Chilli*, acessar a página de autenticação, onde deverá informar o nome e senha corretos.

Após conectar no sistema com sucesso, uma janela *pop up* aparecerá para o usuário, informando que o mesmo está conectado. Além disso, esta tela exibirá também um cronômetro, que marcará o tempo de conexão do cliente. Caso o cliente necessite desconcetar-se da internet, o mesmo deverá efetuar *Logout* e sair do sistema, como mostra a figura 5.6 .

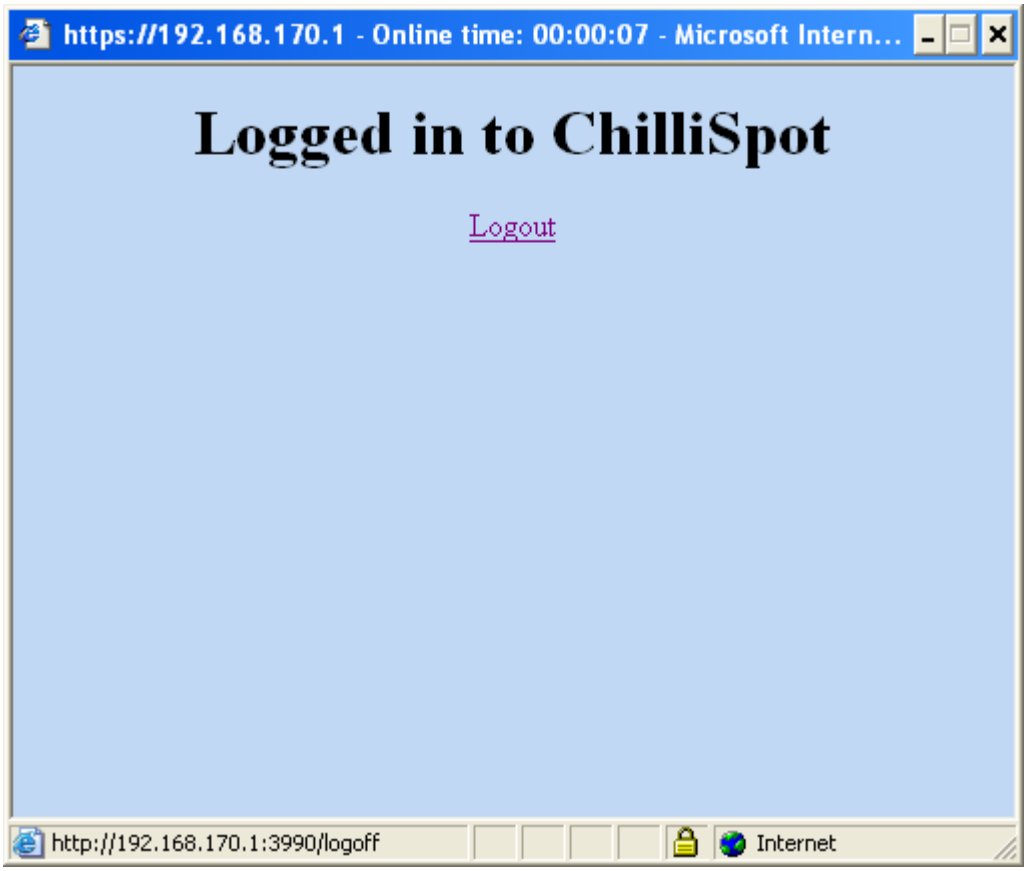

Figura 5.6: Página de desconexão de clientes

Testes de controle de banda foram realizados conforme mostra as figuras 5.7 e 5.8 onde o usuário, configurado com o IP 192.168.190.5, estava limitado a 64k de *download* e 64 de *upload*.

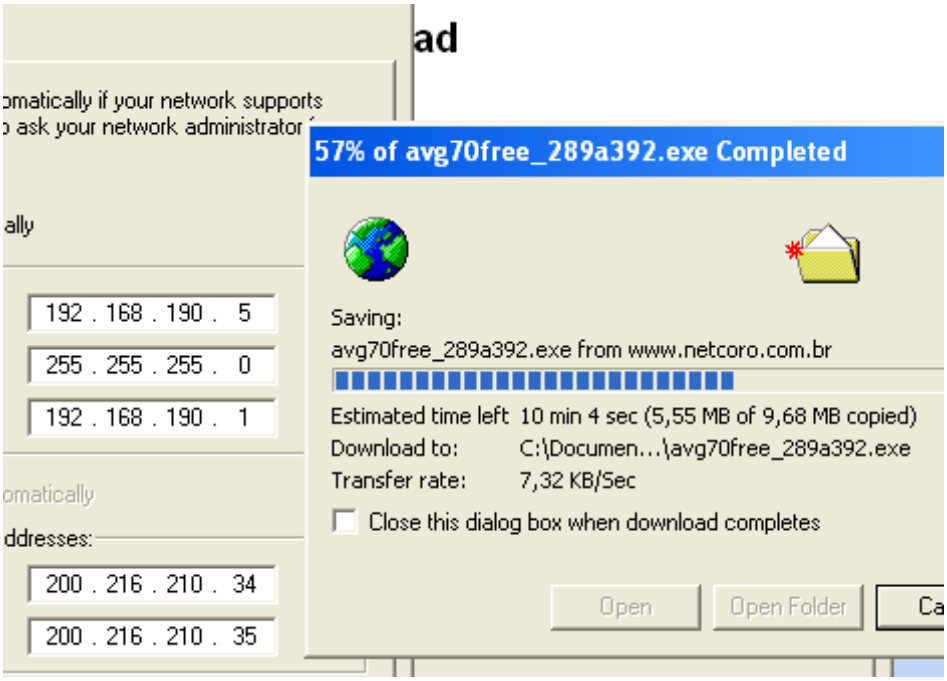

Figura 5.7: Realização de teste de *download* com controle de banda

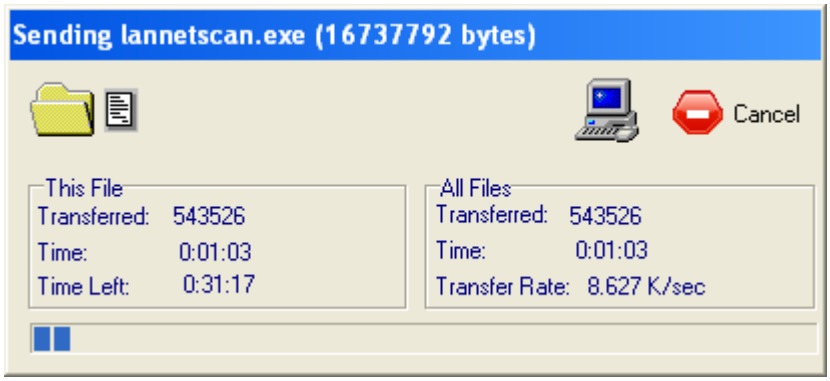

Figura 5.8: Realização de teste de *upload* com controle de banda

Testes em relação a troca de IP para acessar, indevidamente, a banda de outro cliente, foram realizados sem sucesso, portanto, o cliente só conseguirá conectar-se à Internet se estiver configurado com o seu repectivo *IP*, *MacAddress e* senha, caso contrário, o acesso é negado e bloqueado.

Testes em relação à clonagem de *MAC* foram realizados. Nos testes, a máquina clone ao tentar entrar no sistema, estando o cliente clonado conectado, exibirá uma tela informando que o IP está duplicado na rede e o acesso é bloqueado.

Neste sistema, mesmo não eliminando totalmente o risco de clonagem de *MAC* na rede, consegue-se detectar o problema, fato não exibido no sistema proprietário, onde a máquina clone navega sem ser percebida no sistema.

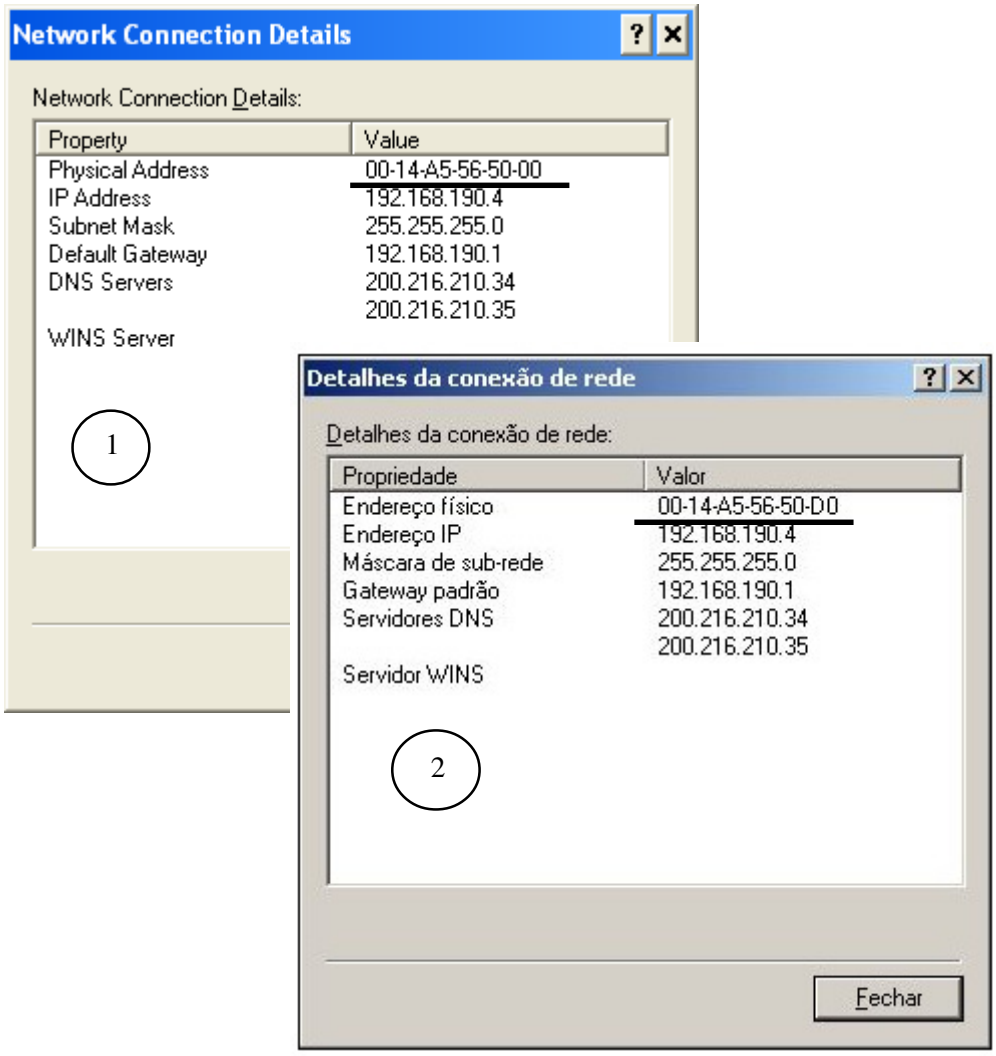

Figura 5.9: Janela com detalhes da exibição das configuração de rede de máquina clonada e máquina clone: 1 - *Notebook* clonado 2 - PC máquina clone.

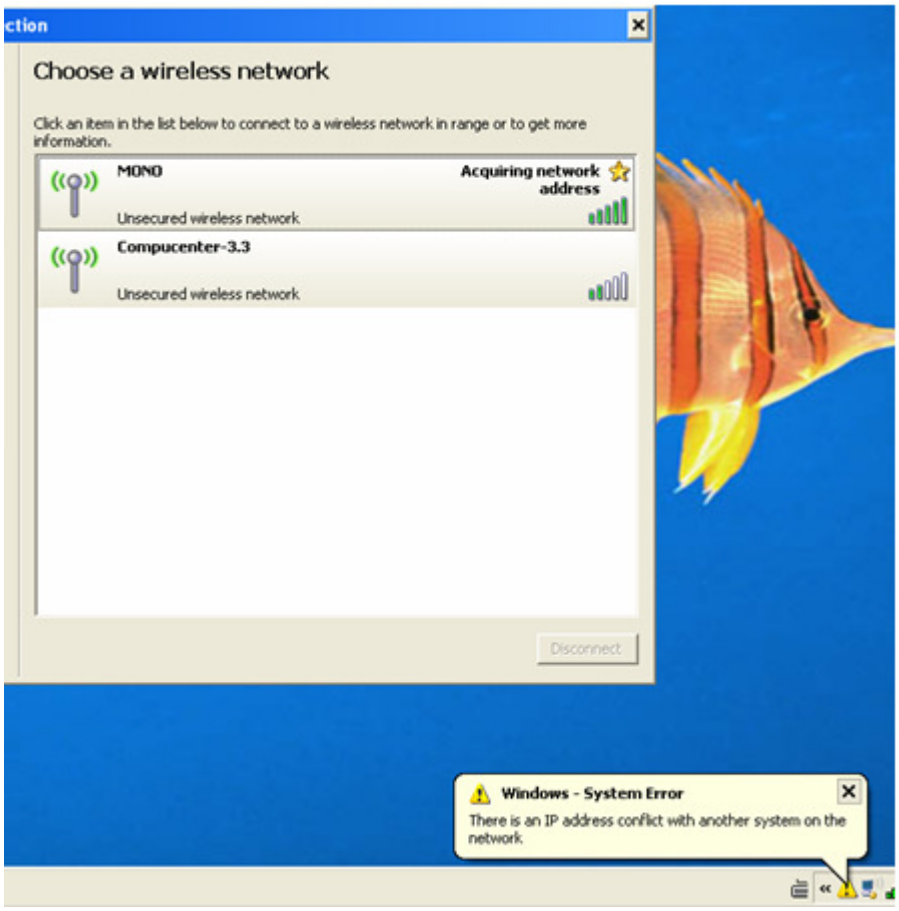

Figura 5.10: Tela do *Notebook* clonado exibindo mensagem de conflito de IP na rede

5.3 Gerenciamento

A seguir, serão exibidas algumas figuras com telas do programa *PhpRADmin*, onde foram cadastrados os usuários. Serão exibidas, também, algumas figuras com gerenciamento de usuários conectados e alguns relatórios que podem ser gerados pelo sistema.

Com o *PhpRADmin,* além de gerenciamento de clientes, existem outros controles como: controle de usários do sistema, controle de *backup* da base de dados, geração de certificados, etc.

Figura 5.11: Tela de *Login* do programa *PhpRADmin*

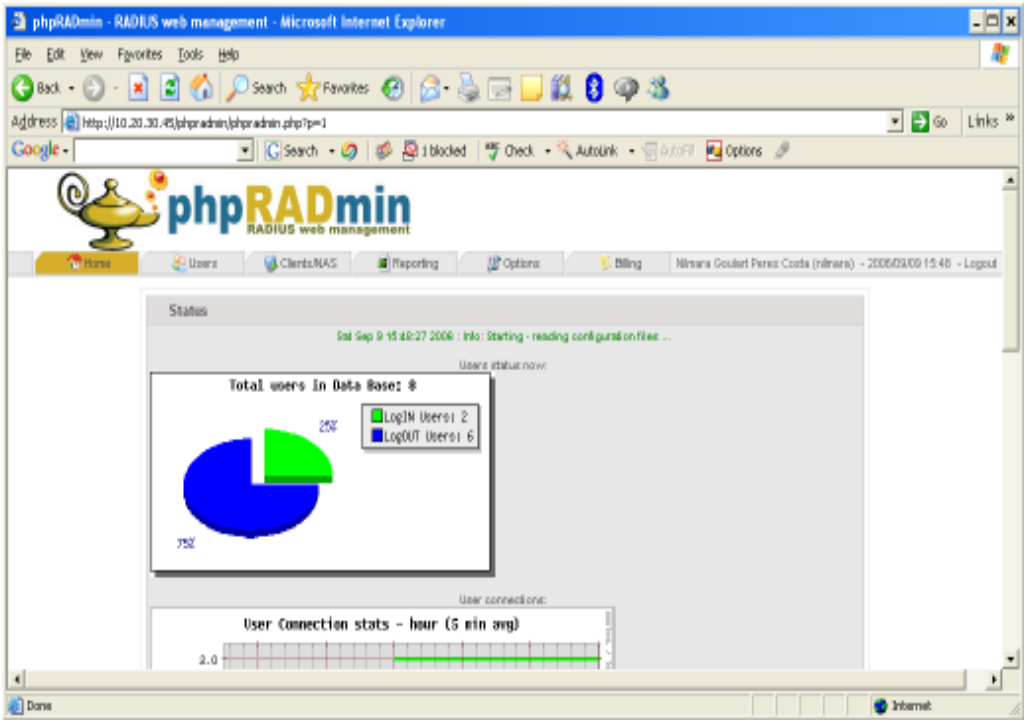

Figura 5.12: Tela de Gerenciamento do *PhpRADmin*

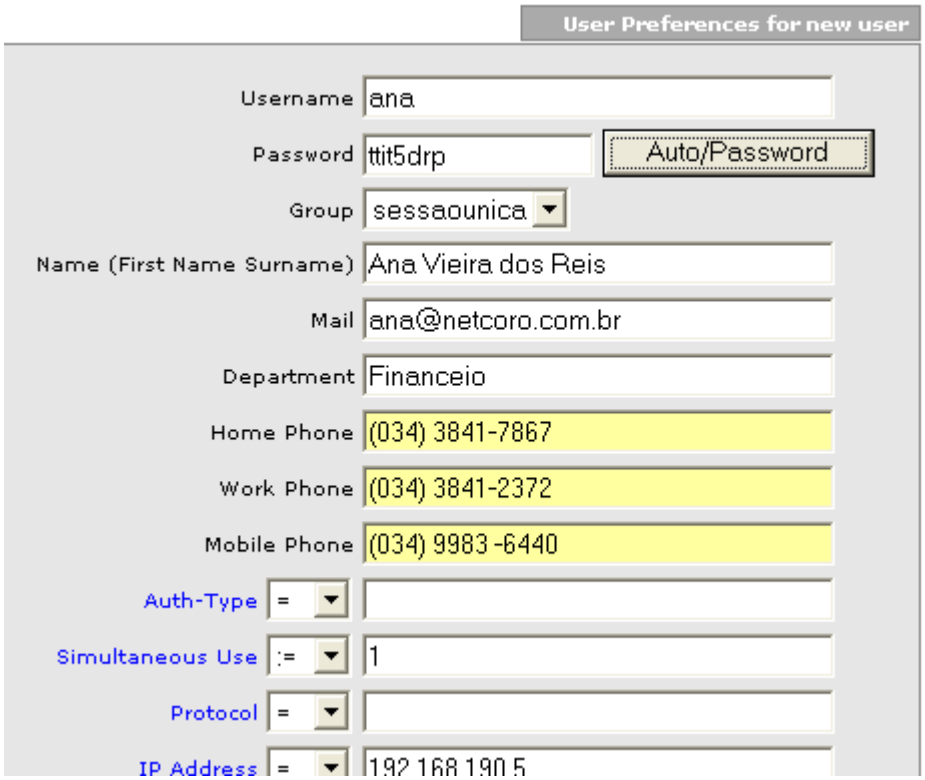

Figura 5.13: Detalhes da tela de cadastro de usuários do *PhpRADmin*

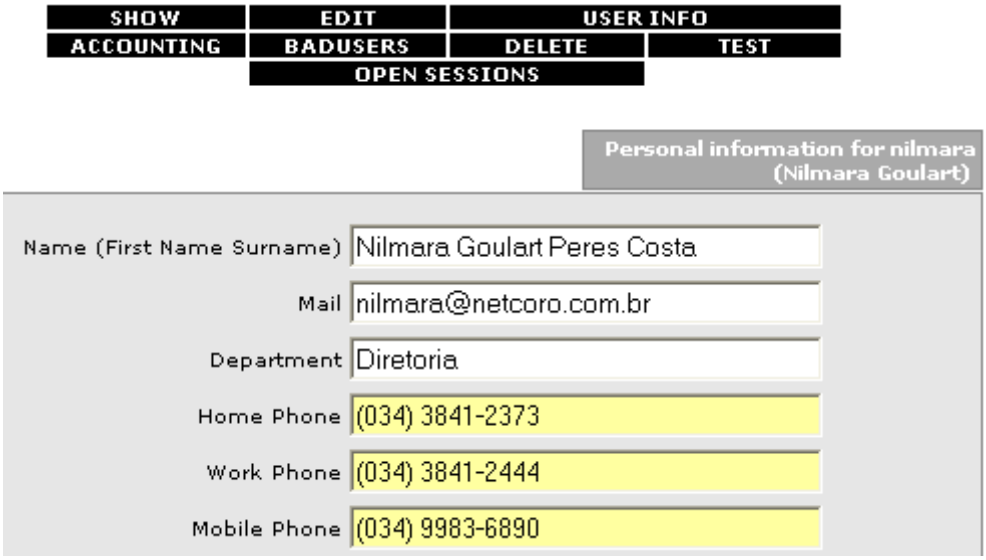

Figura 5.14: Detalhes da tela de gerenciamento de usuários do *PhpRADmin*

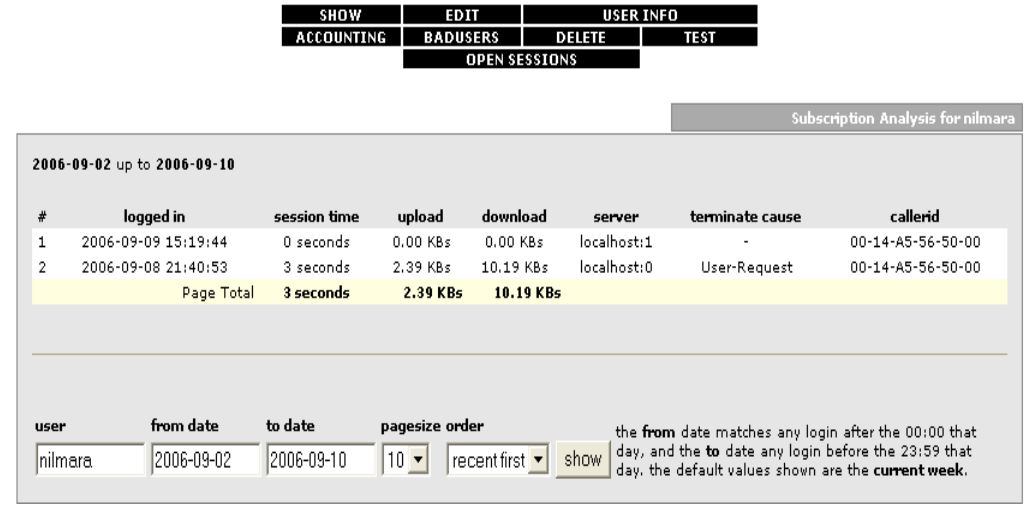

Figura 5.15: Detalhes da tela de contabilização de *upload/download* de usuários

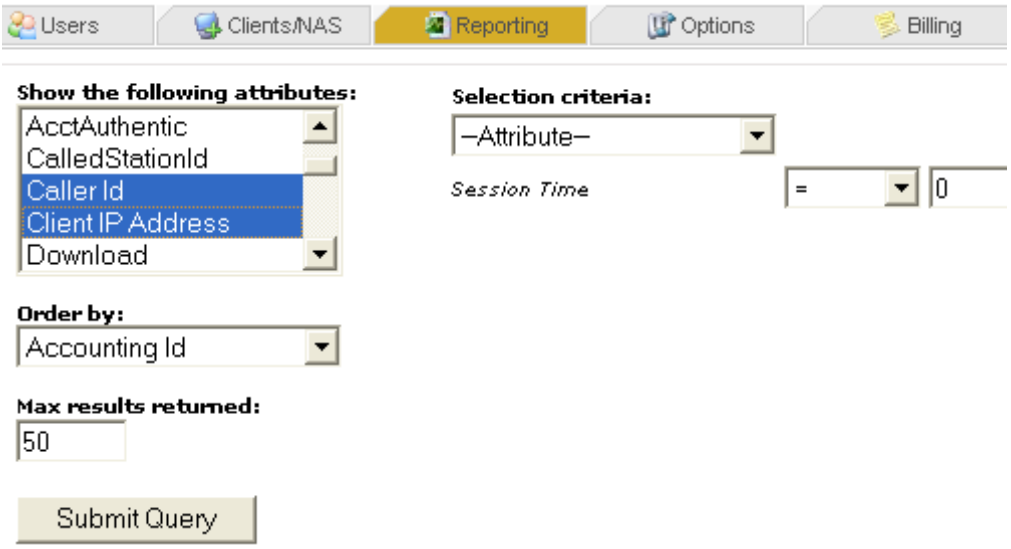

Figura 5.16 Detalhes da tela de geração de relatórios com escolha de atributos

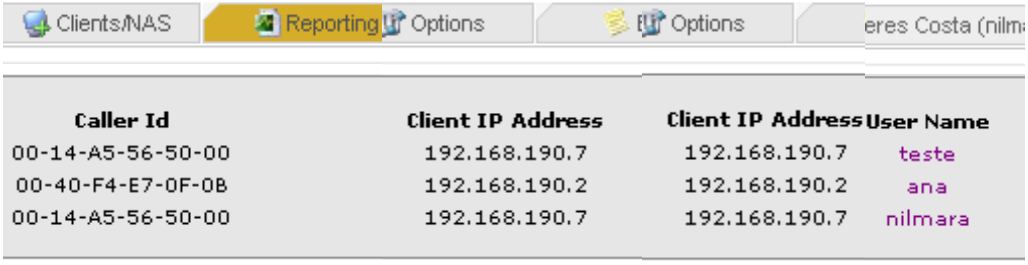

Figura 5.17: Detalhes do relatório de usuários online com escolha de atributos

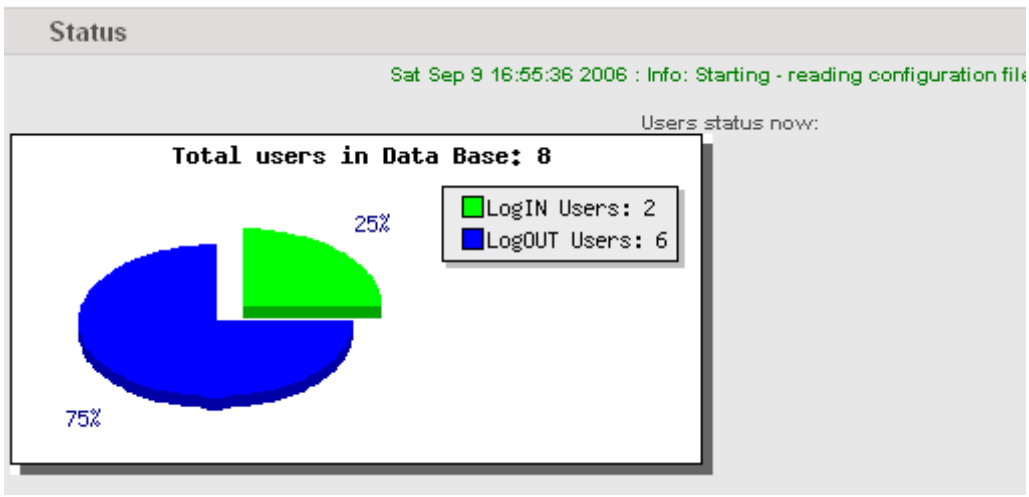

Figura 5.18: Detalhes de um dos gráfico gerados pelo *PhpRADmin*

Figura 5.19: Detalhes da tela desconexão de usuários *online*

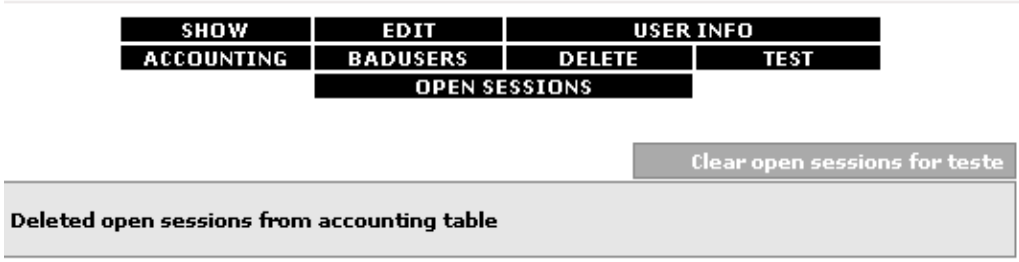

Figura 5.20: Detalhes da tela após a desconexão de usuários *online*

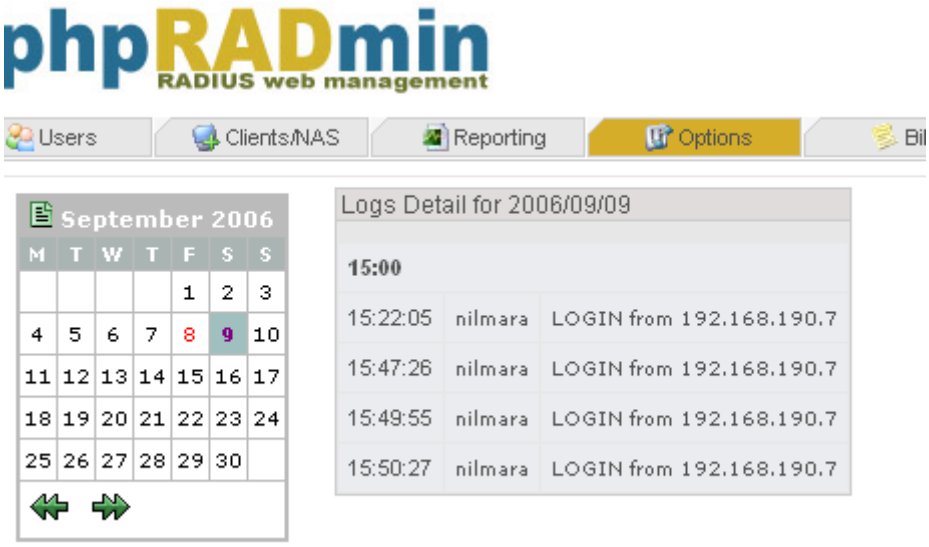

Figura 5.21: Detalhes da tela de *login* de usuário no *PhpRADmin*

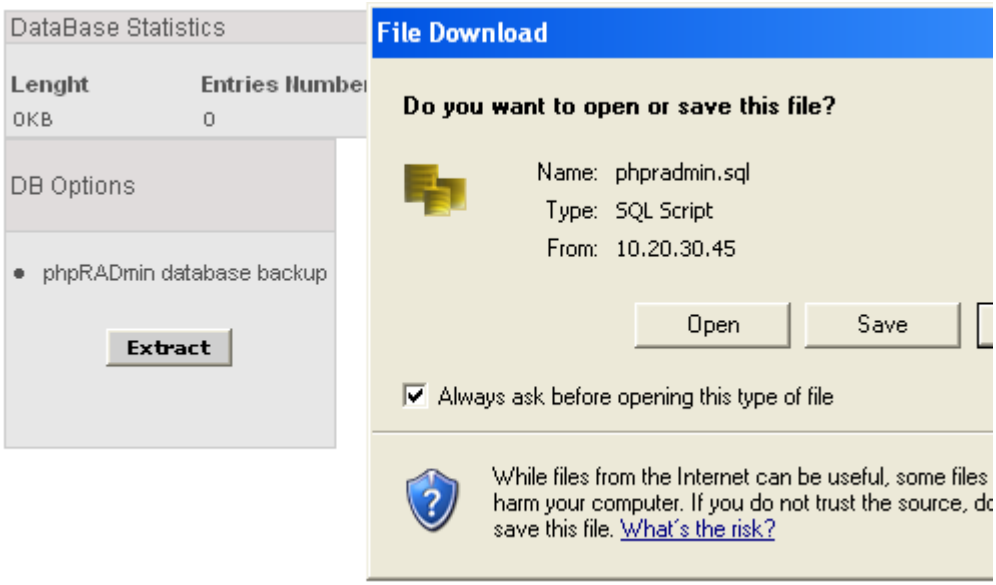

Figura 5.21: Detalhes da tela de criação de *backup* no *PhpRADmin*

```
Last login: Sun Sep 10 09:46:45 2006
Linux 2.4.32.
root@mono:~# tail -f /var/log/radius.log
Sun Sep 10 10:35:41 2006 : Auth: Login OK: [teste/123] (from client localhost po rt 0 cli 00-14-A5-56-50-00)
Sun Sep 10 11:09:03 2006 : Auth: Login OK: [teste/123] (from client localhost po rt 0 cli 00-14-A5-56-50-00)
Sun Sep 10 13:29:21 2006 : Auth: Login OK: [teste/123] (from client localhost po rt 0 cli 00-14-A5-56-50-00)
Sun Sep 10 17:17:03 2006 : Auth: Multiple logins (max 1) [MPP attempt]: [ana/123 ] (from client localhost port 1 cli 00-40-F4-E7-
0F-0B)
Sun Sep 10 17:17:10 2006 : Auth: Multiple logins (max 1) [MPP attempt]: [ana/123 ] (from client localhost port 1 cli 00-40-F4-E7-
0F-0B)
Sun Sep 10 17:17:17 2006 : Error: Invalid operator for item Framed-IP-Address: r everting to '=='
Sun Sep 10 17:17:17 2006 : Info: rlm_sql (sql): No matching entry in the databas e for request from user [vitor]
Sun Sep 10 17:17:17 2006 : Auth: Login incorrect: [vitor/123] (from client local host port 1 cli 00-40-F4-E7-0F-0B)
Sun Sep 10 17:17:27 2006 : Auth: Login incorrect: [teste/13434] (from client loc alhost port 1 cli 00-40-F4-E7-0F-0B)
Sun Sep 10 17:17:34 2006 : Auth: Login OK: [nilmara/nil2372] (from client localh ost port 1 cli 00-40-F4-E7-0F-0B)
```

Figura 5.22: Arquivo de *log* com conexões de usuários bem/mal sucedidas

Capítulo 6

Conclusão e Propostas de Continuidade

Chegou-se a conclusão, que é possível utilizar o sistema para conexão de usuários *wireless* à Internet, utilizando-se de programas gratuitos que podem, satisfatoriamente, substituir o sistema atual ou ainda funcionar como sistema de *backup* no caso de danificação de equipamentos, com as vantagens abaixo relacionadas.

- baixo custo;
- maior confiabilidade no sistema;
- maior gerenciamento de usuários;
- melhor gerenciamento de pacotes na rede;

Como o processamento do sistema é realizado em um computador e não em um *Acess Point*, a performace de funcionamento, quando da utilização do sistema por vários usuários, será superior à performance da *Acess Point*.

A princípio, o projeto tinha como objetivo a redução de custos com equipamentos, *softwares* e assistências técnica, mas, além disso, o APlinux mostrou-se mais seguro que o sistema atualmente implantado. Outras vantagens foram constatadas, como um maior controle sobre os usuários conectados, pelo fato do servidor Radius permitir armazenamento de várias informações de conexão de usuários, como: tentativas sem sucesso de conexões, relatórios de horas e acompanhamento de conexão de usuários, etc.

Outros aspectos do sistema ainda devem ser refinados, como teste de perfomance, melhoria na *interface* de cadastro e controle de usuários e ainda submeter o sistema a teste mais criteriosos de segurança.

A parte gráfica do sistema requer melhoramentos, como a criação de *interfaces* de autenticação personalizada com a logomarca do Provedor de Acesso, etc.

Os próximos testes, a serem realizados, serão em um ponto de acesso com poucos usuários, para monitoração do novo sistema e detecção de problemas que poderão ocorrer.

Para um bom funcionamento de um sistema de internet via *Wireless,* é necessário a combinação de vários fatores, tais como: análise do ambiente, implementação de criptografia, forte autenticação e constante monitoração.

A proposta de continuidade, está na implementação de um sistema que poderá ser desenvolvido com ferramenta *PHP*, que além de automatizar todas as tarefas de gerenciamento, possa agregar mais recursos e informações sobre usuários. Estes recursos, seriam, dentre outros, um sistema integrado envolvendo controle de pagamento de usuários, sistema de cobrança, sistema de suporte técnico e sistema de vendas .

Implementar um controle de banda mais eficiente, utilizando a ferramenta *HTB*, como foi mencionado no Capítulo 5, visando melhoria de *QOS*.

Com relação a segurança da rede, diversos aspectos devem ser considerados no desenvolvimento de trabalhos futuros, que poderão implementar um sistema de segurança eficaz, com ferramentas de monitoração, sistema de armazenamento de senhas, etc, envolvendo os usuários, pontos de acesso e servidores.

Enfim, um projeto focado na segurança das informações transmitidas na rede *wireless*, e gereciamento dos pontos de acesso, com o intuito de minimizar os riscos e vulnerabilidades comuns e associados à essa tecnologia.

---

[1] *QOS - Quality of Services* - Qualidade de Serviço: Refere-se à garantia de largura de banda ou, como em muitos casos, é utilizada informalmente para referir a probabilidade de um pacote circular entre dois pontos de rede.

7 Referências Bibliográficas

AirStrike: **Uma Implementação de Segurança para Redes IEEE 802.11b**. Acesso em julho de 2006. Disponível em: <*http://www.airstrike.ravel.ufrj.br*>.

Application Layer Packet Classifier for Linux. 2006. Acesso em julho de 2006. Disponível em : <*http://l7-filter.sourceforge.net*>.

BELTRAME, Michele *HOWTO Chillispot with FreeRadius and MySQL*. 2006. Acesso em agosto de 2006. Disponível em : < *http://gentoo-wiki.com/HOWTO_Chillispot_with_FreeRadius_and_MySQL*>.

BRANDÃO, Patrick. Freeradius - **Servidor radius eficiente e completo**. 2005 . Acesso em julho de 2006. Disponível em : <*http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=1842*>.

CERT.br . **Cartilha de Segurança para Internet Versão 3.0** . 2006. Acesso em julho de 2006. Disponível em : <*http://cartilha.cert.br/*>.

Installation de chillispot sur une Debian sarge 2004 . Acesso em julho de 2006 . Disponível em : <*http://www.pervasive-network.org/SPIP/Installation-de-chillispot-sur-une*>.

JAKOBSEN, Jens. *Chillispot.*2004. Acesso em julho de 2006. Disponível em: <*http://www.chillispot.org/*>.

Manual do *MySQL*. Acesso em julho de 2006. Disponível em <*http://dev.mysql.com/doc/refman/5.0/en/index.html*>.

Nocat. Acesso em julho de 2006. Disponível em <*http://nocat.net/*>.

Oasis. Ferramenta para autenticação de usuários. Página visitada em em julho 2006. Disponível em <*http://www.stockholmopen.net* >.

RUFINO, Nelson Murilo de O. **Segurança em Redes sem Fio. Aprenda a proteger suas informações em ambienes Wi-Fi e Bluetooth** . São Paulo: Novatec Editora Ltda, 2005 224p.

SANCHES, Carlos Alberto. **Projetando Redes *WLAN*: conceitos e práticas**. São Paulo: Èrica, 2005 341 p.

SMITH, Roderick W. **Redes Linux Avançadas**. Rio de Janeiro : Editora Ciência Moderna Ltda, 2003 630p.

SOARES, Walace. **PHP 5 Conceitos, Programação e Integração com Banco de Dados**. São Paulo: Érica, 2004 523 p.

SOUZA, de Antonio Sérgio. **Instalação *Slackware* 10.2 Linux com suporte a *IPP2P*, *Layer7*, *Time*, *CONNMARK* e *CLASSIFY***. Página visita em em julho 2006. Disponível em <*http://www.uailinux.com.br/slackwareinstall.html*>.

The phpMyAdmin Project Effective MySQL Management. Acesso em julho de 2006. Disponível em : *http://www.phpmyadmin.net.*

The Slackware Linux Project . Acesso em julho de 2006. Disponível em <*http:// www.slackware.com/*>.

VALIÑAS, González Manuel. *Seguridad en Redes 802.11x. 2006*. Acesso em julho de 2006. Disponível em <*http://www.atc.uniovi.es/inf_med_gijon/3iccp/2006/trabajos/ wifi/*>

Wireless HotSpot HowTo. Acesso em agosto de 2006. Disponível em <*http:// www.howtoforge.com/wireless_hotspot_howto*> .