

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RN
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES**

THALES EDUARDO BERNARDO DE FIGUEIREDO

**IMPLANTAÇÃO DE HOTSPOT NO TRIBUNAL REGIONAL DO TRABALHO DA
21ª REGIÃO - TRT21**

**NATAL/RN
2010**

THALES EDUARDO BERNARDO DE FIGUEIREDO

**IMPLANTAÇÃO DE HOTSPOT NO TRIBUNAL REGIONAL DO TRABALHO DA
21ª REGIÃO - TRT21**

**Monografia apresentada ao curso de
Especialização em Redes de
Computadores da Faculdade Natalense
para o Desenvolvimento do Rio Grande
do Norte como requisito parcial para a
obtenção do título de Especialista em
Redes de Computadores.**

**Orientador: Aluizio Ferreira da Rocha
Neto (Mestre)**

**NATAL/RN
2010**

RESUMO

Este trabalho tem por finalidade a implantação de um Hotspot para disponibilizar acesso a internet sem fio aos servidores públicos, ao escritório da OAB e aos advogados que vão para as audiências no Tribunal Regional do Trabalho da 21ª Região - TRT21. Para a implantação do Hotspot foram utilizadas algumas ferramentas como o Nocat que é a ferramenta fundamental do projeto, o FreeRadius que faz a checagem das informações, o OpenLDAP que armazena todas as informações cadastradas, o DHCP que fornece o endereçamento para o computador cadastrado, o Squid que faz o controle de acesso a endereços da internet, o Sarg que gera relatórios de acesso a internet por usuário, o MRTG que gera gráficos do link de internet e para completar o conjunto das ferramentas utilizadas o Apache, Bind, Snmp e Ntp. Para que os usuários tenham acesso a internet sem fio através do Hotspot, terá que ser feito um cadastro prévio fornecendo alguns dados. Com os dados cadastrados, os usuários poderão ter acesso a internet sem fio através da autenticação no Hotspot.

Palavra-Chave: HotSpot, Wireless, Nocat, Linux, internet.

ABSTRACT

This work aims at implementing a Hotspot to provide wireless Internet access to public servants, the OAB's office and the lawyers who go to hearings on the Regional Labor Court of 21 ° Region - TRT21. Hotspot location for some tools were used as NoCat which is the fundamental tool of the project, FreeRadius which checks the information, OpenLDAP that stores all the information registered, the DHCP server provides the address registered for the computer, the Squid that controls access to web addresses, Sarg generate reports that Internet access by user, MRTG generates graphs and internet link to complete the set of tools used Apache, Bind, NTP and SNMP. For users to access the Internet wirelessly through the Hotspot, has to be made a previous registration by providing some data. With the data registered, users can access the Internet wirelessly through the Hotspot authentication.

Keyword: HotSpot, Wireless, Nocat, Linux, internet.

SUMÁRIO

1 INTRODUÇÃO.....	8
1.1 PROBLEMA DE PESQUISA.....	8
1.2 OBJETIVOS.....	8
1.2.1 OBJETIVOS ESPECÍFICOS.....	8
1.3 JUSTIFICATIVA.....	9
2 REFERENCIAL TEORICO.....	10
2.1 PROCESSO DE VALIDAÇÃO.....	10
2.2 FERRAMENTAS NECESSÁRIAS.....	11
2.2.1 NOCAT.....	11
2.2.2 FREERADIUS.....	11
2.2.3 OPENLDAP.....	11
2.2.4 APACHE.....	12
2.2.5 BIND.....	12
2.2.6 DHCP.....	12
2.2.7 SQUID	13
2.2.8 MRTG.....	13
2.2.9 SNMP.....	13
2.2.10 SARG.....	13
2.2.11 NTP.....	14
3 DESENVOLVIMENTO DA SOLUÇÃO.....	15
3.1 INSTALAÇÃO E CONFIGURAÇÃO DO LDAP.....	16
3.2 INSTALAÇÃO E CONFIGURAÇÃO DO DHCP3-SERVER-LDAP.....	18
3.3 INSTALAÇÃO E CONFIGURAÇÃO DO SERVIDOR RADIUS.....	22
3.4 INSTALAÇÃO E CONFIGURAÇÃO DO NOCAT.....	26
3.4.1 CONFIGURANDO O NOCAT GATEWAY.....	29
3.4.2 CONFIGURANDO O NOCAT AUTHSERV.....	29
3.4.3 CONFIGURANDO O FIREWALL DO NOCAT.....	30
3.5 INSTALAÇÃO E CONFIGURAÇÃO DO APACHE.....	37
3.6 INSTALAÇÃO E CONFIGURAÇÃO DO DNS (BIND).....	42
3.7 INSTALAÇÃO E CONFIGURAÇÃO DO PROXY (SQUID)	44
3.8 INSTALAÇÃO E CONFIGURAÇÃO DO NTP.....	49
3.9 INSTALAÇÃO E CONFIGURAÇÃO DO SARG.....	50
3.10 INSTALAÇÃO E CONFIGURAÇÃO DO MRTG COM SNMP.....	52
3.11 PORTAL DE CADASTRO DE USUÁRIOS.....	55
4 CONCLUSÃO.....	59
REFERÊNCIAS.....	60

LISTA DE FIGURAS

Figura 1: Processo de Validação.....	10
Figura 2: Esquema da Rede Wireless.....	15
Figura 3: Sistema e Equipamentos Utilizados.....	15
Figura 4: Comando que atualiza a lista de pacotes.....	16
Figura 5: Comando que atualiza o Sistema Operacional.....	16
Figura 6: Exemplo de configuração do arquivo interfaces.....	16
Figura 7: Comando que instala o Servidor LDAP.....	17
Figura 8: Exemplo de configuração do arquivo slapd.conf.....	17
Figura 9: Exemplo de configuração do arquivo ldap.conf.....	18
Figura 10: Comando que reinicia o serviço do LDAP.....	18
Figura 11: Comando que instala o Servidor DHCP.....	18
Figura 12: Exemplo de configuração do arquivo dhcp3-server.....	19
Figura 13: Comando que faz a copia do arquivo dhcp.schema.gz para a pasta schema.....	19
Figura 14: Comando que faz a copia do arquivo dhcp.schema.gz para a pasta schema.....	19
Figura 15: Exemplo de configuração adicionadas no arquivo slapd.conf.....	19
Figura 16: Exemplo de configuração do arquivo dhcpd.conf.....	20
Figura 17: Exemplo de configuração do arquivo dhcp.ldif.....	21
Figura 18: Comandos que para o serviço do ldap e remove os arquivos.....	21
Figura 19: Comandos que adiciona as configurações do dhcp no servidor ldap e dar permissões.....	21
Figura 20: Comando que inicia o servidor ldap.....	22
Figura 21: Comando que instala os pacotes dependentes do Servidor Radius.....	22
Figura 22: Comando que descompacta o arquivo freeradius.....	22
Figura 23: Comando que faz a instalação do Servidor Radius.....	23
Figura 24: Exemplo de configuração realizada no arquivo clients.conf.....	23
Figura 25: Exemplo de configuração realizada no arquivo ldap.....	23
Figura 26: Opção que será habilitada na seção authorize do arquivo default.....	23
Figura 27: Opção que será habilitada na seção authentication do arquivo default.....	24
Figura 28: Opção padrão do arquivo radius.conf.....	24
Figura 29: Opção após alteração do arquivo radius.conf.....	24
Figura 30: Exemplo do script do radius.....	25
Figura 31: Comandos que fazem com que o radius seja iniciado com o sistema.....	25
Figura 32: Comandos que fazem a instalação do módulo de autenticação.....	25
Figura 33: Comandos que reinicia o servidor Radius.....	25
Figura 34: Comandos que inicia o servidor Radius em modo debug.....	26
Figura 35: Comandos que cria link das lib's.....	26
Figura 36: Opção de configuração que será adicionada no arquivo slapd.conf.....	26
Figura 37: Comandos que reinicia o servidor LDAP.....	26
Figura 38: Comando que descompacta o arquivo nocat.....	27
Figura 39: Opção de configuração padrão.....	27
Figura 40: Opção de configuração após alteração.....	27
Figura 41: Comando que cria um diretório chamado nocat.....	27
Figura 42: Comando que instala o módulo gateway do Nocat.....	27
Figura 43: Comando que instala o módulo de autenticação do Nocat.....	27
Figura 44: Comando que remove as chaves existentes por padrão.....	28
Figura 45: Comando que cria uma nova chave.....	28
Figura 46: Exemplo de opções para criar uma nova chave.....	28

Figura 47: Comandos que copia a chave para as pastas pgp.....	28
Figura 48: Comando que altera os donos das pastas pgp.....	29
Figura 49: Comando que altera as permissões das pastas pgp.....	29
Figura 50: Exemplo de configuração do arquivo nocat.conf do módulo gateway.....	29
Figura 51: Exemplo de configuração do arquivo nocat.conf do módulo authserv.....	30
Figura 52: Opção de configuração padrão do arquivo radius.pm.....	30
Figura 53: Opção de configuração após alteração do arquivo radius.pm.....	30
Figura 54: Exemplo de configuração do arquivo de Firewall chamado extra.fw.....	33
Figura 55: Exemplo de configuração para habilitar o script extra.fw.....	34
Figura 56: Exemplo de configuração da regra padrão do arquivo initialize.fw.....	34
Figura 57: Exemplo de configuração da regra após alteração do arquivo initialize.fw.....	34
Figura 58: Exemplo de configuração do script reset.fw.....	34
Figura 59: Exemplo de configuração do script rotacionarlog.sh.....	35
Figura 60: Exemplo de configuração para habilitar o script rotacionarlog.sh no Cron.....	36
Figura 61: Exemplo de configuração do script nocat.sh.....	36
Figura 62: Comandos necessários para colocar o script nocat.sh na inicialização.....	36
Figura 63: Portal de Autenticação para acesso a internet.....	37
Figura 64: Comando que instala o Servidor Apache.....	37
Figura 65: Comando que habilita o módulo ssl no apache.....	37
Figura 66: Comando que cria um certificado auto-assinado.....	38
Figura 67: Exemplo de configuração para habilitar o módulo ssl nos sites desejados.....	38
Figura 68: Exemplo de configuração do VirtualHost Portal.....	39
Figura 69: Comando que irá criar o arquivo .httpasswd.....	39
Figura 70: Exemplo de configuração do VirtualHost Proxy.....	40
Figura 71: Exemplo de configuração do VirtualHost Nocat.....	41
Figura 72: Comandos que cria o certificado auto-assinado do nocat.....	41
Figura 73: Comandos que habilitam os Virtualhost's criados.....	41
Figura 74: Exemplo de configuração das portas do apache no arquivo ports.conf.....	42
Figura 75: Comando que reinicia o apache.....	42
Figura 76: Comando que instala o Servidor DNS.....	42
Figura 77: Exemplo de configuração de zona.....	42
Figura 78: Exemplo de configuração de domínio DNS.....	43
Figura 79: Exemplo de configuração do arquivo named.conf.options.....	43
Figura 80: Exemplo de configuração do arquivo resolv.conf.....	44
Figura 81: Comando que reinicia o Servidor DNS.....	44
Figura 82: Comando que instala o Servidor Proxy.....	44
Figura 83: Exemplo de configuração adicionadas no arquivo squid.conf.....	45
Figura 84: Exemplo de configuração do arquivo dominios_bloq. ACL para domínios bloqueados.....	46
Figura 85: Exemplo de configuração do arquivo dominios_ok. ACL para domínios permitidos	46
Figura 86: Exemplo de configuração do arquivo downloads_ok. ACL para downloads permitidos.....	46
Figura 87: Exemplo de configuração do arquivo palavras_bloq. ACL para palavras bloqueadas	46
Figura 88: Comando que reinicia o Servidor Proxy.....	46
Figura 89: Exemplo de configuração do arquivo wpad.dat.....	47
Figura 90: Exemplo de configuração adicionada ao DHCP para o WPAD.....	47
Figura 91: Exemplo de configuração adicionada ao DNS para o WPAD.....	47
Figura 92: Exemplo de configuração adicionada ao Proxy para o WPAD.....	47

Figura 93: Exemplo de configuração do proxy no cliente.....	48
Figura 94: Exemplo de configuração do código padrão do arquivo initialize.fw.....	48
Figura 95: Exemplo de configuração do código após alteração do arquivo initialize.fw.....	48
Figura 96: Comando que reinicia o Nocat.....	49
Figura 97: Exemplo de erro após autenticação no Nocat.....	49
Figura 98: Exemplo do código padrão do arquivo gateway.pm.....	49
Figura 99: Exemplo do código após alteração do arquivo gateway.pm.....	49
Figura 100: Comando que instala o NTP.....	50
Figura 101: Exemplo de configuração de servidores públicos no arquivo ntp.conf.....	50
Figura 102: Comando que reinicia o NTP.....	50
Figura 103: Comando que instala o SARG.....	50
Figura 104: Exemplo de configuração a ser realizada no arquivo sarg.conf.....	51
Figura 105: Comandos que fazem a remoção dos scripts do Sarg.....	51
Figura 106: Exemplo do script sarg-dia.sh.....	51
Figura 107: Comando que dar permissão de execução ao script sarg-dia.sh.....	51
Figura 108: Exemplo de configuração adicionada ao crontab para execução do script sarg-dia.sh.....	51
Figura 109: Relatório gerado pelo Sarg.....	52
Figura 110: Comando que instala o MRTG e o SNMP.....	52
Figura 111: Exemplo de configuração da comunidade habilitada no arquivo snmpd.conf.....	53
Figura 112: Comando que reinicia o SNMP.....	53
Figura 113: Comando que verifica a conexão com o SNMP.....	53
Figura 114: Comando que gera as configurações do MRTG.....	53
Figura 115: Exemplo de configuração de um link de 2 Megas.....	53
Figura 116: Comando que cria a página de acesso aos gráficos do MRTG.....	54
Figura 117: Gráficos gerados pelo MRTG.....	54
Figura 118: Tela inicial do Portal de Cadastro.....	55
Figura 119: Tela de Cadastro dos usuários.....	56
Figura 120: Tela de Cadastro de administradores.....	57
Figura 121: Tela de Consulta de usuários.....	57
Figura 122: Tela de Consulta e/ou Cadastrado de equipamento dos usuários.....	58

1 INTRODUÇÃO

Hotspot é uma rede *wireless* configurada para acesso compartilhado à Internet. O *host* do Hotspot adquire um ponto de acesso wireless, conecta o dispositivo à Internet e transmite seu sinal dentro de um local público. Qualquer pessoa que tenha uma placa wireless dentro do alcance do ponto de acesso do host poderá acessar a rede e usar a Internet.

Colocando um pouco de segurança, pode-se evitar que qualquer pessoa se conecte na rede. Para isso existem varias formas de se evitar e uma delas será descrita neste trabalho que é através de um portal de autenticação.

O usuário que for utilizar o Hotspot terá que fazer um prévio cadastro com o administrador, para que seu computador possa ter acesso a rede sem fio. Com o computador cadastrado na rede sem fio, o usuário ao abrir seu browser será redirecionado para um portal de autenticação, onde fornecerá o login e senha que foi cadastrado previamente. Após isso, o usuário terá acesso a internet porém com todas as conexões controladas e filtradas através de um proxy (Squid) e também de um firewall (iptables).

Todos os acessos a internet serão registrados e poderão ser visualizados pelo administrador através dos relatórios gerados diariamente pelo Sarg.

1.1 PROBLEMA DE PESQUISA

O problema de pesquisa abordado por este trabalho foi estudar o funcionamento das ferramentas Nocat, Radius e LDAP, para poder ser feito a integração entre elas e o serviço do portal de autenticação funcionar.

1.2 OBJETIVOS

O objetivo desse projeto é disponibilizar acesso a internet sem fio através de um ponto de acesso chamado de Hotspot.

1.2.1 Objetivos Específicos

Para atingir o objetivo geral, as seguintes etapas serão seguidas:

- Configurar um servidor Linux com algumas ferramentas para que seja feito a autenticação dos usuários;
- Permitir acesso à internet somente a usuários autenticados;
- A autenticação terá que ser feita em uma página web e a base de dados deverá ser armazenada em um base de dados LDAP;
- Usar HTTPS na página de autenticação;

1.3 JUSTIFICATIVA

A motivação para criação deste trabalho foi a necessidade que o Tribunal Regional do Trabalho da 21ª Região — TRT21, tinha em disponibilizar acesso a internet para o escritório da OAB, que fica instalado em uma sala no próprio Tribunal, para que os advogados que estivessem nas audiências pudessem fazer as consultas dos processos e qualquer outro tipo de trabalho, mas que esse acesso não comprometesse a segurança da rede interna, como por exemplo computadores infectados de vírus e infectando toda a rede do Tribunal.

Além disso, o link disponibilizado pode ser usado como um link de backup ou também para testes de acesso externo a internet do próprio Tribunal.

2 REFERENCIAL TEORICO

2.1 PROCESSO DE VALIDAÇÃO

O Processo de Validação do Hotspot funciona da seguinte forma:

O usuário faz o pedido de acesso aos recursos da rede, com isso o DHCP recebe a informação de que um computador com Endereço MAC específico está solicitando um endereçamento IP, então o DHCP faz uma busca do MAC no LDAP, caso o MAC do computador não estiver cadastrado, o acesso é rejeitado e não é atribuído IP. Caso o MAC estiver cadastrado, o DHCP fornecerá um IP para o computador.

Quando o usuário abrir o browser, será redirecionado para o site de autenticação. Neste momento o usuário fornece as credenciais para ter acesso a internet, então o FreeRadius coleta essas credenciais e faz uma busca no LDAP, caso as credenciais estiverem erradas ou não cadastradas, o acesso será rejeitado, caso estejam válidas o acesso será liberado e o usuário poderá acessar a internet.

A Figura 1 mostra o processo de validação do HotSpot:

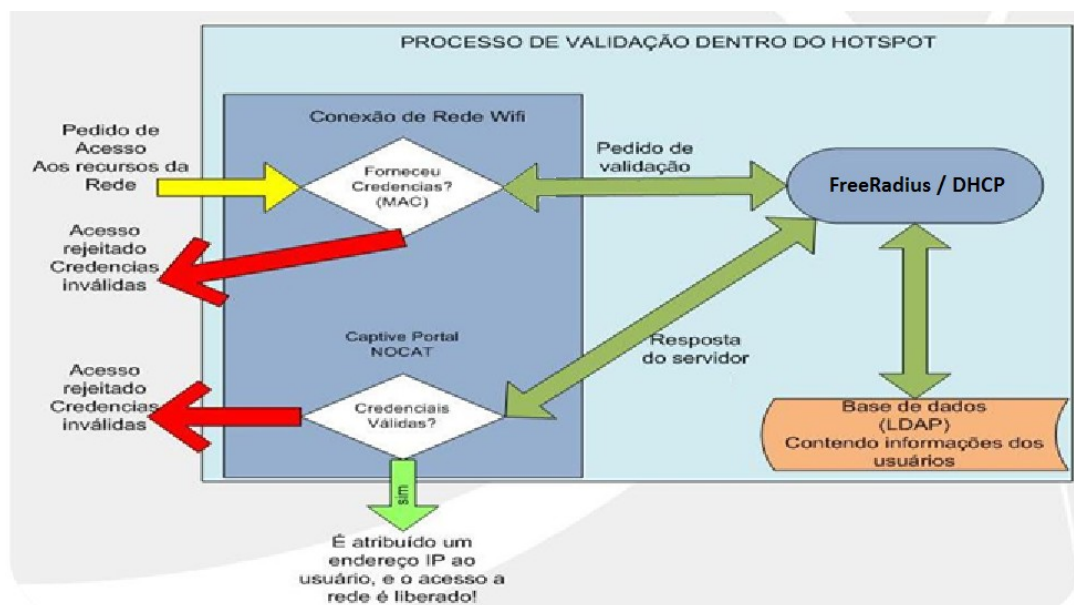


Figura 1: Processo de Validação

Fonte: Criação Própria

2.2 FERRAMENTAS NECESSÁRIAS

2.2.1 Nocat

O programa NoCatAuth, criado pela nocat.net tem o propósito de evitar que pessoas não autorizadas tenham acesso a internet em rede wireless. Para isso, esse programa em Perl captura o tráfego na porta 80 e o desvia para um servidor HTTP, onde haverá uma página solicitando usuário e senha, e opcionalmente, uma página para cadastro.

O NoCat também pode ser usado para simplesmente obrigar o usuário a ler uma página de *splash* antes de navegar, onde se pode colocar um termo de uso, um aviso ou simplesmente a página do provedor de acesso.

2.2.2 FreeRadius

O FreeRadius é um servidor Radius *open source*. Radius é um protocolo amplamente empregado para disponibilizar acesso a redes com autenticação, autorização e contabilização. Ele foi desenvolvido, originalmente para uso em serviços de acesso discado, mas, pela sua simplicidade, eficiência e facilidade de implementação, hoje é suportado por servidores de VPN e acesso a redes sem fio.

2.2.3 OpenLDAP

O OpenLDAP foi desenvolvido a partir do LDAP 3.3 da Universidade de Michigan. Seu projeto é gerenciado pela OpenLDAP Foundation.

O LDAP (*Lightweight Directory Access Protocol*) é um protocolo (executado sobre o TCP/IP) cliente-servidor, utilizado para acessar um serviço de Diretório. Ele foi inicialmente usado como uma interface para o X.500, mas também pode ser usado com autonomia e com outros tipos de servidores de Diretório. Atualmente vem se tornando um padrão, diversos programas já têm suporte a LDAP.

2.2.4 Apache

O Apache é um servidor Web extremamente configurável, robusto e de alta performance desenvolvido por uma equipe de voluntários (conhecida como Apache Group) buscando criar um servidor web com muitas características e com código fonte disponível gratuitamente via Internet. Segundo a Netcraft, o Apache é mais usado que todos os outros servidores web do mundo juntos.

2.2.5 Bind

O BIND (*Berkeley Internet Name Domain* ou, como chamado previamente, *Berkeley Internet Name Daemon*) é o servidor para o protocolo DNS mais utilizado na Internet, especialmente em sistemas do tipo Unix, onde ele pode ser considerado um padrão. Foi criado por quatro estudantes de graduação, membros de um grupo de pesquisas em ciência da computação da Universidade de Berkeley, e foi distribuído pela primeira vez com o sistema operacional 4.3BSD. O programador Paul Vixie, enquanto trabalhava para a empresa DEC, foi o primeiro mantenedor do BIND. Atualmente o BIND é suportado e mantido pelo Internet Systems Consortium.

Para a versão 9, o BIND foi praticamente reescrito. Ele passou a suportar, dentre outras funcionalidades, a extensão DNSSEC e os protocolos TSIG e IPv6.

2.2.6 Dhcp

O DHCP (*Dynamic Host Configuration Protocol* ou protocolo de configuração dinâmica de endereços de rede) é um protocolo de serviço TCP/IP que permite que todos os micros da rede recebam suas configurações de rede automaticamente a partir de um servidor central, sem que você precise ficar configurando os endereços manualmente em cada um. Este protocolo é o sucessor do BOOTP que, embora mais simples, tornou-se limitado para as exigências atuais.

2.2.7 Squid

O Squid é um servidor proxy cache de alto desempenho para clientes web. Desenvolvido em código-aberto sob a licença GNU GPL, é o resultado da contribuição de milhares de programadores do mundo inteiro. Completo, estável e seguro; apesar de normalmente associado ao universo do Linux, roda em vários outros sistemas operacionais, incluindo o Windows. Embora seja usado principalmente para protocolos HTTP e FTP, tem suporte para outros como TLS, SSL e HTTPS.

2.2.8 Mrtg

O MRTG (*Multi Router Traffic Grapher*) é um programa poderoso para monitoramento. Ele cria gráficos com os dados recolhidos, de forma que possam visualizar dados de forma mais prática do que a numérica.

2.2.9 Snmp

O SNMP (*Simple Network Management Protocol* - Protocolo Simples de Gerência de Rede) é um protocolo de gerência definido a nível de aplicação, é utilizado para obter informações de servidores SNMP - agentes espalhados em uma rede baseada na pilha de protocolos TCP/IP. Os dados são obtidos através de requisições de um gerente a um ou mais agentes utilizando os serviços do protocolo de transporte UDP para enviar e receber suas mensagens através da rede.

2.2.10 Sarg

O Sarg (*Squid Analysis Report Generator*) é uma ferramenta que gera relatórios a partir dos logs de acesso do Squid permitindo verificar os sites da internet acessados pelos usuários.

2.2.11 Ntp

O NTP (*Network Time Protocol*) é um protocolo para sincronização dos relógios dos computadores, ou seja, ele define um jeito para um grupo de computadores conversar entre si e acertar seus relógios, baseados em alguma fonte confiável de tempo, como os relógios atômicos do Observatório Nacional, que definem a Hora Legal Brasileira. Com o NTP é fácil manter o relógio do computador sempre com a hora certa, com exatidão de alguns milésimos de segundo.

3 DESENVOLVIMENTO DA SOLUÇÃO

A Figura 2 mostra topologia da solução a ser utilizada como estudo de caso para análise das soluções de softwares aqui estudados.

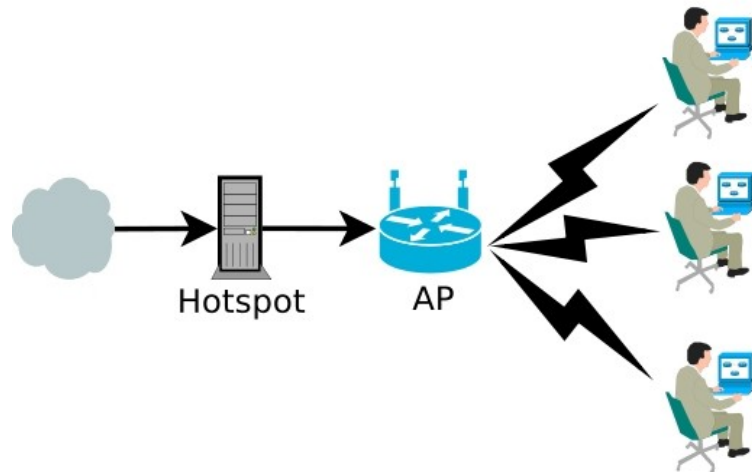


Figura 2: Esquema da Rede Wireless

Fonte: Criação Própria

Para a implementação dessa solução, foram utilizados os seguintes itens de hardware:

- Computador ITAUTEC com processador Intel(R) Xeon(TM) CPU 2.80GHz, 1Gb de RAM com Debian 5.0.3 Lenny e 2 placas de rede;
- Access Point 3com Wireless 7760 11a/b/g PoE.



Figura 3: Sistema e Equipamentos Utilizados

Fonte: google.com

A primeira coisa a ser feita é a atualização da lista de pacotes do apt-get, para permitir

que ele busque os pacotes mais atuais, quando solicitado. A Figura 4 mostra o comando para atualizar a lista de pacotes:

```
# apt-get update
```

Figura 4: Comando que atualiza a lista de pacotes

Fonte: Criação Própria

A Figura 5 mostra o comando para atualizar o Sistema Operacional:

```
# apt-get dist-upgrade
```

Figura 5: Comando que atualiza o Sistema Operacional

Fonte: Criação Própria

Após a atualização do Sistema Operacional, serão configuradas as interfaces de rede. O arquivo a ser editado será o *interfaces*, localizado em */etc/network*.

A Figura 6 mostra um exemplo de configuração:

```
# Interface eth0 — Internet
auto eth0
iface eth0 inet static
    address 189.45.111.178
    netmask 255.255.255.248
    network 189.45.111.176
    broadcast 189.45.111.183
    gateway 189.45.111.177
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 192.168.2.1 189.45.64.10 189.45.112.2

# Interface eth1 — intranet
auto eth1
    iface eth1 inet static
    address 192.168.2.1
    network 192.168.2.0
    netmask 255.255.255.0
    broadcast 192.168.2.255
```

Figura 6: Exemplo de configuração do arquivo interfaces

Fonte: Criação Própria

3.1 INSTALAÇÃO E CONFIGURAÇÃO DO LDAP

A instalação do Servidor LDAP é necessária, pois é onde ficarão armazenadas as

informações dos clientes da rede wireless. O OPENLDAP foi selecionado para tal função. Sua versão é a 2.4.11-1+lenny1 para a implementação deste trabalho. A Figura 7 mostra o comando para instalar o Servidor LDAP:

```
# aptitude install slapd ldap-utils libldap2-dev
```

Figura 7: Comando que instala o Servidor LDAP

Fonte: Criação Própria

Para configurar é necessário entrar na pasta `/etc/ldap/` e editar os arquivos `slapd.conf` e `ldap.conf`. A Figura 8 e 9 mostra exemplos de configurações que devem ser acrescentadas e/ou habilitadas nestes arquivos:

```
# Arquivo slapd.conf

moduleload      back_bdb
backend         bdb
database        bdb
suffix          "dc=wifi,dc=trt21,dc=jus,dc=br"          # -> Sufixo do domínio.

access to attrs=userPassword,shadowLastChange
    by dn="cn=admin,dc=wifi,dc=trt21,dc=jus,dc=br" write
    by anonymous auth
    by self write
    by * none

access to dn.base="" by * read

access to *
    by dn="cn=admin,dc=wifi,dc=trt21,dc=jus,dc=br" write
    by * read
```

Figura 8: Exemplo de configuração do arquivo slapd.conf

Fonte: Criação Própria

```
# Arquivo ldap.conf:

BASE      dc=wifi,dc=trt21,dc=jus,dc=br
SCOPE      sub
TIMELIMIT  30
SIZELIMIT  0

# A URI para o servidor LDAP. No caso, roda no proprio servidor(localhost)
URI ldap://127.0.0.1/

# Define a versão para os comando LDAP. O default é usar a versão 3 (se suportada pelos
clientes)
ldap_version 3

# Define o atributo correspondente ao login name do usuário
pam_login_attribute uid

# Filtro a ser utilizado conjuntamente com o <pam_login_attribute>=%s.
pam_filter objectclass=posixAccount
```

Figura 9: Exemplo de configuração do arquivo ldap.conf

Fonte: Criação Própria

Após a configuração, reiniciar o serviço. A Figura 10 mostra o comando para reiniciar o serviço do LDAP:

```
# /etc/init.d/sldapd restart
```

Figura 10: Comando que reinicia o serviço do LDAP

Fonte: Criação Própria

3.2 INSTALAÇÃO E CONFIGURAÇÃO DO DHCP3-SERVER-LDAP

A instalação do Servidor DHCP é necessária, pois é onde ficarão armazenadas as informações de endereçamento IP dos clientes da rede wireless. O DHCP3-SERVER-LDAP foi selecionado para tal função. Sua versão é a 3.1.1-6+lenny4 para a implementação deste trabalho. A Figura 11 mostra o comando para instalar o Servidor DHCP:

```
# aptitude install dhcp3-server-ldap
```

Figura 11: Comando que instala o Servidor DHCP

Fonte: Criação Própria

Depois de instalado deve-se configurar aonde o dhcp irá fornecer o endereçamento. Então é necessário entrar na pasta `/etc/default/` e editar o arquivo `dhpc3-server`, colocando a

interface de rede interna. A Figura 12 mostra um exemplo de configuração do arquivo *dhpc3-server*:

```
INTERFACES="eth1"
```

Figura 12: Exemplo de configuração do arquivo *dhpc3-server*

Fonte: Criação Própria

Para configurar o servidor dhcp usando como base de dados o servidor ldap é necessário fazer a configuração dos arquivos *slapd.conf*, *dhcp.conf* e *dhcp.schema*. Segue abaixo os procedimentos:

Copiar o arquivo *dhcp.schema.gz* para a pasta schema do ldap. A Figura 13 mostra o comando que faz a copia do arquivo:

```
# cp /usr/share/doc/dhcp3-server-ldap/dhcp.schema.gz /etc/ldap/schema
```

Figura 13: Comando que faz a copia do arquivo *dhcp.schema.gz* para a pasta schema

Fonte: Criação Própria

Entra na pasta */etc/ldap/schema* e descompactar o arquivo *dhcp.schema.gz*. A Figura 14 mostra os comandos para descompactar o arquivo:

```
# cd /etc/ldap/schema
# gunzip dhcp.schema.gz
```

Figura 14: Comando que faz a copia do arquivo *dhcp.schema.gz* para a pasta schema

Fonte: Criação Própria

Entrar na pasta */etc/ldap/* e editar o arquivo *slapd.conf* acrescentando algumas configurações. A Figura 15 mostra um exemplo de configuração que devem ser adicionadas no arquivo:

```
Include /etc/ldap/schema/dhcp.schema
index      cn                      eq
index      dhcpHWAddress          eq
index      dhcpClassData          eq
```

Figura 15: Exemplo de configuração adicionadas no arquivo *slapd.conf*

Fonte: Criação Própria

Entrar na pasta */etc/dhcp3/* e editar o arquivo *dhcpd.conf* acrescentando algumas configurações. A Figura 16 mostra um exemplo de configuração:

```
Ldap-server "127.0.0.1";
ldap-port 389;
ldap-username "cn=admin,dc=wifi,dc=trt21,dc=jus,dc=br";
ldap-password "123456";      # SENHA DO LDAP
ldap-base-dn "ou=dhcp,dc=wifi,dc=trt21,dc=jus,dc=br";
ldap-dhcp-server-cn "hotspot"; # NOME DO SERVIDOR
ldap-method dynamic;
ldap-debug-file "/var/log/dhcp-ldap-start.log";
```

Figura 16: Exemplo de configuração do arquivo dhcpd.conf

Fonte: Criação Própria

Para criar as configurações do dhcp no ldap é necessário criar um arquivo chamado *dhcp.ldif* e colocar algumas configurações. A Figura 17 mostra um exemplo das configurações:

```
# Cria uma OU para agrupar os usuarios
dn: ou=Usuarios,dc=wifi,dc=trt21,dc=jus,dc=br
objectClass: organizationalUnit
ou: Usuarios

# Cria uma OU para agrupar os objetos relacionados ao DHCP
dn: ou=dhcp,dc=wifi,dc=trt21,dc=jus,dc=br
objectClass: organizationalUnit
ou: dhcp

# Especifica o nome do host do servidor LDAP
dn: cn=hotspot,ou=dhcp,dc=wifi,dc=trt21,dc=jus,dc=br
objectClass: top
objectClass: dhcpServer
cn: hotspot
dhcpServiceDN: cn=dhcpserv,ou=dhcp,dc=wifi,dc=trt21,dc=jus,dc=br

# Árvore de configuração que aponta para Servidor.
dn: cn=dhcpserv,ou=dhcp,dc=wifi,dc=trt21,dc=jus,dc=br
cn: dhcpserv
objectClass: top
objectClass: dhcpService
dhcpPrimaryDN: cn=hotspot,ou=dhcp,dc=wifi,dc=trt21,dc=jus,dc=br
dhcpStatements: authoritative
dhcpStatements: ddns-update-style none
dhcpStatements: default-lease-time 3600
dhcpStatements: max-lease-time 86400
dhcpStatements: option wpad-url code 252 = text

# Cria a Sub-rede
dn: cn=192.168.2.0,cn=dhcpserv,ou=dhcp,dc=wifi,dc=trt21,dc=jus,dc=br
cn: 192.168.2.0
objectClass: top
```

```

objectClass: dhcpSubnet
objectClass: dhcpOptions
dhcpOption: domain-name-servers 192.168.2.1,189.45.64.10,189.45.112.2
dhcpOption: routers 192.168.2.1
dhcpOption: subnet-mask 255.255.255.0
dhcpOption: broadcast-address 192.168.2.255
dhcpOption: wpad-url "http://wpad.wifi.trt21.jus.br/wpad.dat\n"
dhcpOption: time-offset -3
dhcpOption: domain-name "wifi.trt21.jus.br."
dhcpNetMask: 24

# Configura um pool para esta sub-rede. Apenas máquinas conhecidas se
# conectarão.
dn:cn=pool,cn=192.168.2.0,cn=dhcpserve,ou=dhcp,dc=wifi,dc=trt21,dc=jus,dc=br
cn: pool
objectClass: top
objectClass: dhcpPool
dhcpRange: 192.168.2.16 192.168.2.254
dhcpPermitList: deny unknown-clients

# Grupo dos Macs Cadastrados
dn: cn=macsok,cn=dhcpserve,ou=dhcp,dc=wifi,dc=trt21,dc=jus,dc=br
objectClass: top
objectClass: dhcpGroup

```

Figura 17: Exemplo de configuração do arquivo dhcp.ldif

Fonte: Criação Própria

Antes de adicionar as informações do dhcp no ldap, faz necessário parar o serviço do Ldap e remover os arquivos de dados. A Figura 18 mostra os comandos que para o serviço do ldap e remove os arquivos:

```

# etc/init.d/slaped stop
# rm -rf /var/lib/ldap/*

```

Figura 18: Comandos que para o serviço do ldap e remove os arquivos

Fonte: Criação Própria

Após ter realizado os comandos da Figura 18, deve-se adicionar as configurações do dhcp no ldap e dar permissão ao usuário e grupo openldap. A Figura 19 mostra os comandos necessário para isso:

```

# slapadd -l dhcp.ldif
# chown -R openldap:openldap /var/lib/ldap

```

Figura 19: Comandos que adiciona as configurações do dhcp no servidor ldap e dar permissões

Fonte: Criação Própria

Iniciar o serviço ldap. A Figura 20 mostra o comando necessário para iniciar o servidor ldap:

```
# /etc/init.d/slapd start
```

Figura 20: Comando que inicia o servidor ldap

Fonte: Criação Própria

3.3 INSTALAÇÃO E CONFIGURAÇÃO DO SERVIDOR RADIUS

A instalação do Servidor Radius é necessária para a autenticação no Servidor LDAP. O freeRadius foi selecionado para tal função. Sua versão atual é a 2.1.8.

Para instalar o Servidor Radius, é necessário antes instalar alguns pacotes:

- build-essential;
- openssl;
- berkleyDB;
- ltdl.h;
- RADIUS-LDAPv3.schema.

A Figura 21 mostra o comando para instalar os pacotes:

```
# aptitude install build-essential openssl libssl-dev libdb4.6 libdb4.6-dev libltdl3 libltdl3-dev
```

Figura 21: Comando que instala os pacotes dependentes do Servidor Radius

Fonte: Criação Própria

Para baixar a ultima versão do freeRadius, é necessário acessar o site <http://freeradius.org/> e efetuar o download da ultima versão. Após download, descompactar o arquivo tgz. A Figura 22 mostra o comando para descompactar o arquivo baixado:

```
# tar -xvzf freeradius-server-2.1.8.tar.gz
```

Figura 22: Comando que descompacta o arquivo freeradius

Fonte: Criação Própria

Para instalar, é preciso entrar na pasta que foi criada e fornecer os comandos para instalação que estão descritos na Figura 23:

```
# ./configure
# make
# make install
```

Figura 23: Comando que faz a instalação do Servidor Radius

Fonte: Criação Própria

Para configurar, é necessário entrar na pasta `/usr/local/etc/raddb/` e efetuar algumas alterações nos arquivos:

- `radius.conf`: localizado na raiz da pasta;
- `clients.conf`: localizado na raiz da pasta;
- `ldap`: localizado na pasta `module`;
- `default`: localizado na pasta `sites-enabled`.

No arquivo `clients.conf`, localizar a opção `secret` que fica dentro de "`client localhost`" e colocar a senha do Servidor Radius como mostra a Figura 24.

```
secret = radiustr21
```

Figura 24: Exemplo de configuração realizada no arquivo clients.conf

Fonte: Criação Própria

Na pasta `modules` a um arquivo chamado `ldap`, configurar algumas opções como mostra a Figura 25:

```
ldap {
    server = "127.0.0.1"
    identity = "cn=admin,dc=wifi,dc=trt21,dc=jus,dc=br"
    password = 123456 # SENHA DO LDAP
    basedn = "ou=Usuarios,dc=wifi,dc=trt21,dc=jus,dc=br"
    filter = "(uid=%{Stripped-User-Name}-%{User-Name})"
    base_filter = "(objectclass=radiusprofile)"
}
```

Figura 25: Exemplo de configuração realizada no arquivo ldap

Fonte: Criação Própria

Na pasta `sites-enabled` a um arquivo chamado `default`, configurar algumas opções como mostra a Figura 26 e Figura 27.

Na seção `Authorize`, descomentar a linha:

```
ldap
```

Figura 26: Opção que será habilitada na seção authorize do arquivo default

Fonte: Criação Própria

Na seção *Authentication*, descomentar as linhas:

```
Auth-Type LDAP {
    ldap
}
```

Figura 27: Opção que será habilitada na seção authentication do arquivo default

Fonte: Criação Própria

Agora editar o arquivo *radius.conf*, deve-se procurar a variável *ipaddr*, a qual será encontrada em dois lugares. A Figura 28 mostra a configuração padrão.

```
ipaddr = *
```

Figura 28: Opção padrão do arquivo radius.conf

Fonte: Criação Própria

A Figura 29 mostra a configuração após alteração, que fará com que o radius aceite somente conexões locais (localhost).

```
ipaddr = 127.0.0.1
```

Figura 29: Opção após alteração do arquivo radius.conf

Fonte: Criação Própria

Para fazer o radius iniciar junto com o sistema, é necessário criar um script para a inicialização, chamado de *radius.sh*. A Figura 30 mostra o conteúdo do script:

```
#Arquivo radius.sh

#!/bin/sh
RADIUSD=/usr/local/sbin/radiusd
test -f $RADIUSD || exit 0
case "$1" in
    start)
        if [ ! -f /var/log/radius.log ]
        then
            :>/var/log/radius.log
        fi
        $RADIUSD
    ;;
    startdebug)
        if [ ! -f /var/log/radius.log ]
        then
            :>/var/log/radius.log
        fi
        $RADIUSD -X
    ;;
    stop)
```

```

        killall radiusd 2>/dev/null
    ;;
restart)
        killall radiusd 2>/dev/null
        sleep 3
        $RADIUSD
    ;;
*)
    echo "Usage: $0 {start|startdebug|stop|restart}"
    exit 1
esac
exit 0

```

Figura 30: Exemplo do script do radius

Fonte: Criação Própria

Depois de criado, dar permissão de execução ao arquivo, coloca-lo no `/etc/init.d` e depois coloca-lo para iniciar com o sistema como mostra a Figura 31:

```

# chmod +x radius.sh
# cp radius.sh /etc/init.d
# update-rc.d radius.sh defaults

```

Figura 31: Comandos que fazem com que o radius seja iniciado com o sistema

Fonte: Criação Própria

Agora é só instalar o módulo de autenticação Perl (Authen::Radius), basta executar o comando descrito na Figura 32 para instalá-lo:

```

# perl -MCPAN -e shell
CPAN1> install Authen::Radius

```

Figura 32: Comandos que fazem a instalação do módulo de autenticação

Fonte: Criação Própria

Depois que instalar o módulo, deve-se reiniciar o radius como descrito na Figura 33:

```

# /etc/init.d/radius.sh restart

```

Figura 33: Comandos que reinicia o servidor Radius

Fonte: Criação Própria

Caso o radius não inicie, deve-se inicia-lo em modo debug e verificar qual erro está acontecendo. A Figura 34 mostra como iniciar o Radius em modo debug:

```
# /etc/init.d/radius.sh startdebug
```

Figura 34: Comandos que inicia o servidor Radius em modo debug

Fonte: Criação Própria

Caso surja erro nas *libfreeradius-radius-2.1.8.so* e *libfreeradius-eap-2.1.8.so* quando iniciar o radius, a solução será fazer um link das lib's. A Figura 35 mostra como fazer os link's das lib's:

```
# ln -s /usr/local/lib/libfreeradius-radius-2.1.8.so /usr/lib/libfreeradius-radius-2.1.8.so
# ln -s /usr/local/lib/libfreeradius-eap-2.1.8.so /usr/lib/libfreeradius-eap-2.1.8.so
```

Figura 35: Comandos que cria link das lib's

Fonte: Criação Própria

Depois que o Radius estiver iniciando corretamente, deve-se editar o arquivo *slapd.conf* localizado na pasta */etc/ldap/*, adicionando a configuração descrita na Figura 36:

```
include /etc/ldap/schema/RADIUS-LDAPv3.schema
```

Figura 36: Opção de configuração que será adicionada no arquivo slapd.conf

Fonte: Criação Própria

Agora o ldap deve ser reiniciado. A Figura 37 mostra o comando que reinicia o servidor LDAP:

```
# /etc/init.d/slaped restart
```

Figura 37: Comandos que reinicia o servidor LDAP

Fonte: Criação Própria

3.4 INSTALAÇÃO E CONFIGURAÇÃO DO NOCAT

A instalação de um Captive Portal é necessária para que haja a captura do primeiro acesso a internet do usuário, fazendo com que ele tenha que passar por uma autenticação. O NOCAT foi selecionado para tal função. Sua versão atual é a 0.82-nightly.

Para baixar a ultima versão do NOCAT, é necessário acessar o site <http://nocat.net> e efetuar o download da ultima versão. Após download, descompactar o arquivo tgz. A Figura 38 mostra o comando para descompactar o arquivo baixado:

```
# tar zxvf NoCatAuth-nightly.tgz
```

Figura 38: Comando que descompacta o arquivo nocat

Fonte: Criação Própria

O Nocat não vem configurado para detectar firewall para kernel 2.6, portanto é preciso alterar manualmente este parâmetro, conforme segue:

Entrar no diretório *bin* (que fica na pasta do nocat) e editar o arquivo *detect-fw.sh*.

A Figura 39 mostra a configuração padrão que terá que ser substituída:

```
test X" `uname -sr | cut -d. -f-2`" = X"Linux 2.4"; then
```

Figura 39: Opção de configuração padrão

Fonte: Criação Própria

A Figura 40 mostra a configuração que terá que ser realizada:

```
test X" `uname -sr | cut -d. -f-2`" = X"Linux 2.6"; then
```

Figura 40: Opção de configuração após alteração

Fonte: Criação Própria

Agora, criar um diretório para o Nocat e executar a instalação. A Figura 41 mostra a criação de um diretório chamado nocat:

```
# mkdir /usr/local/nocat
```

Figura 41: Comando que cria um diretório chamado nocat

Fonte: Criação Própria

Para instalar o módulo gateway, deve ser executado o comando como descrito na Figura 42:

```
# make PREFIX=/usr/local/nocat/gw gateway
```

Figura 42: Comando que instala o módulo gateway do Nocat

Fonte: Criação Própria

Para instalar o módulo de autenticação, deve ser executado o comando como descrito na Figura 43:

```
# make PREFIX=/usr/local/nocat/authserv authserv
```

Figura 43: Comando que instala o módulo de autenticação do Nocat

Fonte: Criação Própria

O Nocat já vem por padrão com uma chave para assinatura digital, então essa chave

deverá ser apagada. A Figura 44 mostra como remover as chaves:

```
# rm -rf /usr/local/nocat/gw/pgp/trustedkeys.pgp
# rm -rf /usr/local/nocat/authserv/pgp/trustedkeys.pgp
```

Figura 44: Comando que remove as chaves existentes por padrão

Fonte: Criação Própria

E agora criar uma nova chave. A Figura 45 mostra como criar uma nova chave:

```
# make PREFIX=/usr/local/nocat/authserv pgpkey
```

Figura 45: Comando que cria uma nova chave

Fonte: Criação Própria

Algumas perguntas serão feitas, responda conforme segue a Figura 46:

```
Escolha DSA (opção 1);
Escolha chave com 1024;
Expira deixe com 0 , confirme nunca expira;
Responda as perguntas de nome, e-mail, conforme necessários;
Confirme suas opções.
```

Figura 46: Exemplo de opções para criar uma nova chave

Fonte: Criação Própria

Após as perguntas descritas na Figura 46, será solicitada uma senha, não deve colocar nada, apenas aperte o botão *enter* e novamente será solicitado para confirmar, *enter* novamente.

Para que seja criado a chave, será pedido que use o computador para que sejam coletados dados aleatórios para a geração da chave. Então, é necessário ficar digitando aleatoriamente, até que em algum instante a chave seja gerada.

Depois de concluído a geração da chave, é necessário copiá-la para as pastas *gw* e *authserv*. A Figura 47 mostra como copiar as chaves para as pastas indicadas:

```
# cp /usr/local/nocat/authserv/trustedkeys.pgp /usr/local/nocat/gw/pgp/
# cp /usr/local/nocat/authserv/trustedkeys.pgp /usr/local/nocat/authserv/pgp/
```

Figura 47: Comandos que copia a chave para as pastas pgp

Fonte: Criação Própria

Para que o servidor Apache possa utilizar as chaves, deve-se alterar o dono destes arquivos. A Figura 48 mostra como alterar os donos das pastas *pgp*:

```
# chown -R www-data:www-data /usr/local/nocat/authserv/pgp
# chown -R www-data:www-data /usr/local/nocat/gw/pgp
```

Figura 48: Comando que altera os donos das pastas pgp

Fonte: Criação Própria

E mudar as permissões, como descrita na Figura 49:

```
# chmod 755 /usr/local/nocat/authserv/pgp
# chmod 755 /usr/local/nocat/gw/pgp
```

Figura 49: Comando que altera as permissões das pastas pgp

Fonte: Criação Própria

3.4.1 Configurando o Nocat Gateway

O Nocat Gateway vai bloquear o acesso a internet e capturar a tela. Para configurar, deve-se editar o arquivo *nocat.conf* que se encontra na pasta */usr/local/nocat/gw*. A Figura 50 mostra um exemplo de configuração do arquivo *nocat.conf*:

```
Verbosity          10
GatewayName        WifiTRT21
GatewayMode        Passive
GatewayLog         /usr/local/nocat/gw/nocat.log
LoginTimeout       600      # 600 é igual a 10 minutos
TrustedGroups      Any
AuthServiceAddr     192.168.2.1
AuthServiceURL      https://$AuthServiceAddr/cgi-bin/login
LogoutURL          https://$AuthServiceAddr/logout.html
ExternalDevice     eth0
InternalDevice     eth1
LocalNetwork       192.168.2.0/24
RouteOnly          1
MembersOnly        1
PGPKeyPath         /usr/local/nocat/gw/pgp
```

Figura 50: Exemplo de configuração do arquivo nocat.conf do módulo gateway

Fonte: Criação Própria

3.4.2 Configurando o Nocat Authserv

O Nocat Authserv, enviará a autenticação para o radius. Para configurar, deve-se editar o arquivo *nocat.conf* que se encontra na pasta */usr/local/nocat/authserv*. A Figura 51 mostra um exemplo de configuração do arquivo *nocat.conf*:

```

Verbosity          10
PGPKeyPath         /usr/local/nocat/authserv/pgp
HomePage           http://www.trt21.jus.br/
DocumentRoot       /usr/local/nocat/authserv/htdocs
DataSource          RADIUS
RADIUS_Host 127.0.0.1:1812
RADIUS_Secret      radiustrt21
RADIUS_TimeOut     5
MinPasswdLength    6
LoginGreeting       Bem Vindo a WIFI do TRT21. Digite usuário e senha para
prosseguir.
LoginMissing        Por favor, preencha todos os campos!
LoginBadUser        Usuario desconhecido. Por favor, tente novamente.
LoginBadPass        Usuario ou Senha desconhecido. Por favor tente novamente.
LoginBadStatus      Desculpe, voce nao esta cadastrado.

```

Figura 51: Exemplo de configuração do arquivo nocat.conf do módulo authserv

Fonte: Criação Própria

Para usar autenticação radius, terá que ser feita uma alteração no arquivo *radius.pm* que fica na pasta */usr/local/nocat/authserv/lib/NoCat/Source*, pois a um bug no módulo.

Por volta da linha 88, irá encontrar o conteúdo descrito na Figura 52:

```

{ Name => 1, Value => $user->id },
{ Name => 2, Value => $user_pw }

```

Figura 52: Opção de configuração padrão do arquivo radius.pm

Fonte: Criação Própria

Terá que ser feito a alteração para o que mostra a Figura 53:

```

{ Name => 1, Value => $user->id, Type => 'string' },
{ Name => 2, Value => $user_pw, Type => 'string' }

```

Figura 53: Opção de configuração após alteração do arquivo radius.pm

Fonte: Criação Própria

3.4.3 Configurando o Firewall do Nocat

Como as regras de firewall do Nocat são muito simples, foi criado um script com novas regras chamado *extra.fw*, segue um exemplo de configuração como mostra a Figura 54:

```
#!/bin/sh
# Arquivo extra.fw
#-----
# Adiciona outras regras as criadas pelo NoCat para o firewall
#-----

# Definindo algumas variaveis..
IF_INT="eth1" # interface interna - REDE WiFi
IF_EXT="eth0" # interface externa - link Internet (ADSL)
IF_LOOP="lo" # interface de loopback
IP_INT="192.168.2.1" # ip interface rede wifi
IP_EXT="189.45.111.178" # ip interface externa
IP_LOOP="127.0.0.1" # ip interface loopback
NET_INT="192.168.2.0/24" # rede interna
NET_EXT="189.45.111.176/29" # rede externa
NET_DMZEXT_TRT21="200.249.4.0/25" # rede DMZ e EXT do TRT21

# C U I D A D O !!!
# As regras serão adicionadas as já criadas pelo NoCat.
# Veja: /usr/local/nocat/gw/initialize.fw

# TABELA: FILTER
# -----
# Definindo as políticas default...
iptables -t filter -P INPUT DROP
iptables -t filter -P FORWARD DROP
iptables -t filter -P OUTPUT ACCEPT
# Cria chains de usuario
# trafego interno (wifi) ->fw
iptables -t filter -N INT-FW 2>/dev/null
iptables -t filter -F INT-FW

# trafego externo->fw
iptables -t filter -N EXT-FW 2>/dev/null
iptables -t filter -F EXT-FW

# trafego liberado pra usuários classe 2 (co-op)
iptables -t filter -N TrafegoC2 2>/dev/null
iptables -t filter -F TrafegoC2

# INPUT
# -----
# Permite os pacotes pertencentes ou relacionados a conexoes previas (TCP, UDP e ICMP)
iptables -A INPUT -p ALL -m state --state ESTABLISHED,RELATED -j ACCEPT

# 1.LOOPBACK
# -----
# Permite todo o trafego para a interface de loopback
iptables -A INPUT -i $IF_LOOP -j ACCEPT

# 2.INT-FW
```

```

# -----
# Desvia trafego chegando pela interface da rede wifi para a chain INT-FW
iptables -A INPUT -i $IF_INT -j INT-FW

# Aceita pacotes direcionados ao proxy (TCP 8080). Squid rodando na mesma maquina...
iptables -A INT-FW -p tcp --dport 8080 -j ACCEPT

# Aceita pacotes DNS (TCP/UDP 53). DNS rodando na mesma maquina...
iptables -A INT-FW -p tcp --dport 53 -j ACCEPT
iptables -A INT-FW -p udp --dport 53 -j ACCEPT

# Aceita pacotes SSH
iptables -A INT-FW -p tcp --dport 22 -j ACCEPT

# Aceita pacotes DHCP. DHCP rodando na mesma maquina...
iptables -A INT-FW -p udp --dport 67:68 -j ACCEPT

# Portas usadas pelo NoCat: HTTP, PORTAL, HTTPS, TCP/5280
iptables -A INT-FW -p tcp -m multiport --dport 80,8181,443,5280 -j ACCEPT

# Rejeita todo o resto
iptables -A INT-FW -j LOG-DROP

# 3.EXT-FW
# -----
# Desvia o trafego chegando pela interface externa (Internet) para a chain EXT-FW
iptables -A INPUT -i $IF_EXT -j EXT-FW
# Aceita pacotes ssh (TCP 22) vindos do TRT
iptables -A EXT-FW -s $NET_DMZEXT_TRT21 -p tcp --dport 22 -j ACCEPT

# Acesso Web - Visualizar o SARG, PORTAL e o MRTG
iptables -A EXT-FW -s $NET_DMZEXT_TRT21 -p tcp -m multiport --dport 443,8181 -j ACCEPT

# Dropa todo o resto
iptables -A EXT-FW -j LOG-DROP

# FORWARD
# -----
# Altera a regra que libera todo o trafego de saida para pacotes marcados com 0x2 redirecionando
pra chain TrafegoC2
iptables -t filter -R NoCat 4 -i $IF_INT -s $NET_INT -m mark --mark 2 -j TrafegoC2

#HTTP, HTTPs, Proxy
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 8080 -j ACCEPT
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 80 -j ACCEPT
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 443 -j ACCEPT
#DNS
iptables -t filter -A TrafegoC2 -p udp -i $IF_INT --dport 53 -j ACCEPT
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 53 -j ACCEPT
#NOCAT

```

```

iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 5820 -j ACCEPT
iptables -t filter -A TrafegoC2 -p udp -i $IF_INT --dport 5820 -j ACCEPT
# POP3 e POP3S
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 110 -j ACCEPT
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 995 -j ACCEPT
# IMAP e IMAPs
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 143 -j ACCEPT
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 993 -j ACCEPT
# SMTP
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 25 -j ACCEPT
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 465 -j ACCEPT
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 587 -j ACCEPT
#FTP
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 20 -j ACCEPT
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 21 -j ACCEPT
#NTP
iptables -t filter -A TrafegoC2 -p udp -i $IF_INT --dport 123 -j ACCEPT
#Portas tcp/udp nao privilegiadas
iptables -t filter -A TrafegoC2 -p tcp -i $IF_INT --dport 1024: -j ACCEPT
iptables -t filter -A TrafegoC2 -p udp -i $IF_INT --dport 1024: -j ACCEPT
#ICMP
iptables -t filter -A TrafegoC2 -p icmp -i $IF_INT -j ACCEPT

# Por fim desabilita todo o resto
iptables -t filter -A TrafegoC2 -i $IF_INT -j LOG-DROP
# PREROUTING
#-----
# Configurando um proxy transparente
# OBS: Essa regra deve estar depois da chain NoCat_Capture
iptables -t nat -A PREROUTING -i $IF_INT -d ! $IP_INT -p tcp --dport 80 -j REDIRECT --to-
port 8080

#
# POSTROUTING
#-----
# O NoCat ja inclui regras pra mascaramento, so que achamos mais interessante criar uma regra
unica SNAT do que MASQUARADE. ATENÇÃO: O parametro "routeonly" (nocat.conf) deve
estar configurado como 1.
iptables -t nat -A POSTROUTING -o $IF_EXT -s $NET_INT -j SNAT --to-source $IP_EXT

```

Figura 54: Exemplo de configuração do arquivo de Firewall chamado *extra.fw*

Fonte: Criação Própria

Com as regras criadas, deve-se editar o arquivo *initialize.fw* que está no diretório */usr/local/nocat/gw/bin/*, para que as regras sejam carregadas. A Figura 55 mostra a configuração que será adicionada no final do arquivo:

```
[ -x /usr/local/nocat/gw/bin/extra.fw ] && /usr/local/nocat/gw/bin/extra.fw
```

Figura 55: Exemplo de configuração para habilitar o script extra.fw

Fonte: Criação Própria

Ainda no arquivo *initialize.fw*, é necessário fazer outra alteração. Essa alteração foi feita pois na inicialização do Nocat, o iptables exibe uma mensagem de erro dizendo que a regra que utiliza DROP está sendo usada erroneamente. Então o que tem que ser feito é substituir o DROP por RETURN.

A Figura 56 mostra a regra padrão:

```
$redirect -m mark --mark $mark -d windowsupdate.microsoft.com -j DROP
```

Figura 56: Exemplo de configuração da regra padrão do arquivo initialize.fw

Fonte: Criação Própria

A Figura 57 mostra a regra após alteração:

```
$redirect -m mark --mark $mark -d windowsupdate.microsoft.com -j RETURN
```

Figura 57: Exemplo de configuração da regra após alteração do arquivo initialize.fw

Fonte: Criação Própria

Outra alteração será no arquivo *reset.fw* que está no diretório */usr/local/nocat/gw/bin/*. As regras da Figura 58 irão zerar todas as outras regras aplicadas anteriormente quando for parar o Nocat. Não esquecer de colocar essas regras, pois caso não seja colocadas, irá dar erro no bind (*rndc: connect failed: 127.0.0.1#953: timed out*) quando for desligar ou reiniciar o servidor.

```
#!/bin/sh
# reset.fw:

# Desabilita o ip forwarding
echo "0" > /proc/sys/net/ipv4/ip_forward
iptables -F
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
iptables -t filter -F
iptables -t filter -X
# Configura politica de INPUT (filter) para ACCEPT;
iptables -t filter -P INPUT ACCEPT
```

Figura 58: Exemplo de configuração do script reset.fw

Fonte: Criação Própria

Uma dica interessante é rotacionar o log do Nocat, pois o Nocat não faz isso automaticamente e com isso o arquivo de log será único e ficará muito grande. Para rotacionar o arquivo de log, deve-se criar um script chamado de *rotacionarlog.sh*. A Figura 59 mostra a configuração do script:

```
#Script rotacionarlog.sh
#!/bin/sh
test -x /etc/init.d/nocat || exit 0
test -e /usr/local/nocat/gw/nocat.log || exit 0
# Pegando a data
DATA=$(date +%d%m%Y)
# Parando o servico
/etc/init.d/nocat stop > /dev/null
# Entrando no diretorio do nocat
cd /usr/local/nocat/gw

# Cria diretorio se nao existir
[ ! -d backup_logs ] && mkdir backup_logs

# Criei um loop onde ele pega o nome de todos
# os arquivos que finalizam com log
# e aplica os comandos abaixo
for ARQUIVO in *log;
do
    # Compactando ARQUIVO
    tar -cf "$ARQUIVO.$DATA.tar" "$ARQUIVO";
    gzip -9 "$ARQUIVO.$DATA.tar";
    mv "$ARQUIVO.$DATA.tar.gz" backup_logs/
    # Rotacionando
    echo "" > "$ARQUIVO"
done
# Iniciando o servico
/etc/init.d/nocat start > /dev/null

# Agora ele apaga os arquivos com mais dias que o especificado na linha debaixo
# Detalhe ele faz essa busca com base na ultima data que o arquivo foi alterado
# e nao na data de criacao do arquivo.
#echo "Apagando logs com mais de 1 ano"
#find /usr/local/nocat/gw/ -mtime +375 -exec rm -f {} \;
```

Figura 59: Exemplo de configuração do script rotacionarlog.sh

Fonte: Criação Própria

Com o script criado, agora é adiciona-lo para ser carregado no crontab. A Figura 60 mostra a configuração que será adicionada ao Cron (Agendador de Tarefas). O script será carregado todos os domingos às 23hrs.

```
00 23 * * 7 root /root/scripts/rotacionarlog.sh
```

Figura 60: Exemplo de configuração para habilitar o script rotacionarlog.sh no Cron

Fonte: Criação Própria

Para colocar o nocat na inicialização do sistema, é necessário criar um script chamado de *nocat.sh*. A Figura 61 mostra a configuração do script:

```
#!/bin/sh
NC=/usr/local/nocat/gw
export PERL5LIB=$NC/lib:$PERL5LIB
export NOCAT=$NC/nocat.conf
case "$1" in
    start)
        echo "Starting the NoCat gateway ..."
        $NC/bin/gateway
        ;;
    stop)
        echo "Stopping the NoCat gateway ..."
        killall gateway 2>/dev/null
        ;;
    restart)
        $0 stop
        sleep 1
        $0 start
        ;;
    *)
        echo "Usage: $0 {start|stop|restart}"
        exit 1
esac
```

Figura 61: Exemplo de configuração do script nocat.sh

Fonte: Criação Própria

Depois de criado, dar permissão de execução ao arquivo, coloca-lo no */etc/init.d* e depois coloca-lo para iniciar com o sistema. A Figura 62 mostra os comandos necessários para a colocar o script *nocat.sh* na inicialização:

```
# chmod +x nocat.sh
# cp nocat.sh /etc/init.d/
# update-rc.d nocat.sh defaults
```

Figura 62: Comandos necessários para colocar o script nocat.sh na inicialização

Fonte: Criação Própria



Figura 63: Portal de Autenticação para acesso a internet

Fonte: Criação Própria

3.5 INSTALAÇÃO E CONFIGURAÇÃO DO APACHE

A instalação do Servidor Web APACHE é necessária, pois é onde ficarão configuradas informações de sites da rede wireless, como o portal de cadastro e o nocat. Sua versão é a 2.2.9-10+lenny7 para a implementação deste trabalho.

A Figura 64 mostra o comando para instalar o Servidor Apache:

```
# apt-get install apache2 apache2-utils
```

Figura 64: Comando que instala o Servidor Apache

Fonte: Criação Própria

Para aumentar a segurança tornando a conexão com os sites mais seguros deve-se habilitar módulo ssl no apache como mostra a Figura 65:

```
# a2enmod ssl
```

Figura 65: Comando que habilita o módulo ssl no apache

Fonte: Criação Própria

Após habilitar o módulo ssl, é necessário criar o certificado. A Figura 66 mostra o comando para criar um certificado auto-assinado:

```
# openssl req $@ -new -x509 -days 365 -nodes -out /etc/apache2/apache.pem -keyout
/etc/apache2/apache.pem
```

Figura 66: Comando que cria um certificado auto-assinado

Fonte: Criação Própria

Para habilitar o módulo ssl nos sites, deve-se adicionar as configurações, como mostra a Figura 67, nos sites que desejam habilitar o módulo ssl.

```
SSLEngine on
ServerSignature On
SSLCertificateFile /etc/apache2/apache.pem
```

Figura 67: Exemplo de configuração para habilitar o módulo ssl nos sites desejados

Fonte: Criação Própria

Neste projeto irá existir 3 (três) Virtualhost:

- Portal: Estará a configuração do MRTG, SARG e Portal de Cadastro;
- Proxy: Estará a configuração do Proxy automático (WPAD).
- Nocat: Estará a configuração do Captive Portal.

Para o Virtualhost Portal, é necessário criar um arquivo no diretório `/etc/apache2/sites-available/` com o nome `portal`. A Figura 68 mostra as configurações do arquivo `portal`.

```
<VirtualHost *:8181>
    Alias /wifitrt /var/www/WifiTRT
    Alias /mrtg /var/www/mrtg
    Alias /sarg /var/www/squid-reports
    ServerName portal.wifi.trt21.jus.br
    DocumentRoot /var/www
    SSLEngine on
    ServerSignature On
    SSLCertificateFile /etc/apache2/apache.pem
    ErrorLog /var/log/apache2/error-portal.log
    CustomLog /var/log/apache2/access-portal.log Combined
    <Directory /var/www/WifiTRT>
        Options +FollowSymLinks
        AllowOverride All
        Order allow,deny
        Allow from all
        DirectoryIndex login.php
    </Directory>
    <Directory /var/www/mrtg>
```

```

        DirectoryIndex index.html
        Options FollowSymLinks Indexes
        AllowOverride AuthConfig
        AuthName "Acesso Restrito"
        AuthType Basic
        AuthUserFile /var/www/.htpasswd
        AuthGroupFile /dev/null
        require valid-user
    </Directory>
    <Directory /var/www/squid-reports>
        Options FollowSymLinks Indexes
        AllowOverride AuthConfig
        AuthName "Acesso Restrito"
        AuthType Basic
        AuthUserFile /var/www/.htpasswd
        AuthGroupFile /dev/null
        require valid-user
    </Directory>
</VirtualHost>

```

Figura 68: Exemplo de configuração do VirtualHost Portal

Fonte: Criação Própria

Como visto na configuração, o site do MRTG e do SARG estão configurados para ter acesso restrito, colocando login e senha antes de visualizar o conteúdo. Para fazer essa configuração deve-se criar um arquivo chamado *.htpasswd* na pasta */var/www* com usuário e senha. A Figura 69 mostra o comando para criar o arquivo *.htpasswd*.

```
# htpasswd -c /var/www/.htpasswd sti
```

Figura 69: Comando que irá criar o arquivo .htpasswd

Fonte: Criação Própria

Para o Virtualhost Proxy, é necessário criar um arquivo no diretório */etc/apache2/sites-available/* com o nome *proxy*. A Figura 70 mostra as configurações do arquivo *proxy*:

```

<VirtualHost wpad.wifi.trt21.jus.br:80>
    DirectoryIndex wpad.dat
    ServerName wpad.wifi.trt21.jus.br
    ServerAlias proxy.* wpad.*
    DocumentRoot "/var/www/proxy"
    ErrorLog /var/log/apache2/error-proxy.log
    CustomLog /var/log/apache2/access-proxy.log combined
    <Directory "/var/www/proxy">
        Order allow,deny
        Allow from 192.168. 127.
        AllowOverride FileInfo AuthConfig Limit
        ForceType application/x-ns-proxy-autoconfig
    </Directory>
    AddType application/x-ns-proxy-autoconfig .dat
    AddType application/x-ns-proxy-autoconfig .pac
    Alias /wpad.dat          "/var/www/proxy/wpad.dat"
    Alias /wpad.da          "/var/www/proxy/wpad.dat"
    Alias /proxy.pac        "/var/www/proxy/wpad.dat"
</VirtualHost>

```

Figura 70: Exemplo de configuração do VirtualHost Proxy

Fonte: Criação Própria

Para o Virtualhost Nocat, é necessário criar um arquivo no diretório */etc/apache2/sites-available/* com o nome *nocat*. A Figura 71 mostra as configurações do arquivo *nocat*:

```

<VirtualHost 192.168.2.1:443>
    DocumentRoot /usr/local/nocat/authserv/htdocs
    ServerName localhost:443
    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/nocat.crt
    SSLCertificateKeyFile /etc/apache2/ssl/nocat.key
    Options FollowSymLinks Includes Indexes MultiViews

    ErrorLog /var/log/apache2/error-nocat.log
    CustomLog /var/log/apache2/access-nocat.log combined
    ScriptAlias /cgi-bin/ /usr/local/nocat/authserv/cgi-bin/

    <Directory "/usr/local/nocat/authserv/htdocs">
        AllowOverride All
        Options FollowSymLinks Includes Indexes MultiViews
        <Limit GET POST OPTIONS PROPFIND>
            Order allow,deny
            Allow from all
        </Limit>
        <LimitExcept GET POST OPTIONS PROPFIND>
            Order deny,allow
            Deny from all
        </LimitExcept>
    </Directory>
</VirtualHost>

```

```

</Directory>
<Directory "/usr/local/nocat/authserv/cgi-bin">
    AllowOverride All
    Options ExecCGI
    <IfModule mod_access.c>
        Order allow,deny
        Allow from all
    </IfModule>

    SetEnv PERL5LIB /usr/local/nocat/authserv/lib
    SetEnv NOCAT /usr/local/nocat/authserv/nocat.conf
    SSLOptions +StdEnvVars
</Directory>
<IfModule mod_setenvif.c>
    SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
</IfModule>

<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteOptions inherit
</IfModule>
</VirtualHost>

```

Figura 71: Exemplo de configuração do VirtualHost Nocat

Fonte: Criação Própria

Além disso, é necessário criar uma chave e um certificado auto-assinado para o VirtualHost Nocat. Para isso, deve-se criar uma pasta chamada *ssl* dentro da pasta do Apache. Entrar na pasta criada e criar os certificados como mostra a Figura 72:

```

# openssl genrsa -out nocat.key 1024
# openssl req -new -key nocat.key -out nocat.csr
# openssl x509 -days 365 -req -in nocat.csr -signkey nocat.key -out nocat.crt

```

Figura 72: Comandos que cria o certificado auto-assinado do nocat

Fonte: Criação Própria

Depois de todos os Virtualhost criados, deve-se habilitá-los. A Figura 73 mostra os comandos para habilitar os Virtualhost's:

```

# a2ensite nocat
# a2ensite proxy
# a2ensite portal

```

Figura 73: Comandos que habilitam os Virtualhost's criados

Fonte: Criação Própria

Após a criação dos Virtualhost's, alterar o arquivo *ports.conf* para configurar as portas. A Figura 74 mostra a configuração das portas do apache:

```
Listen 8181
Listen 192.168.2.1:80
Listen 192.168.2.1:443
```

Figura 74: Exemplo de configuração das portas do apache no arquivo ports.conf

Fonte: Criação Própria

Após configuração, reiniciar o apache como mostra a Figura 75:

```
# /etc/init.d/apache2 restart
```

Figura 75: Comando que reinicia o apache

Fonte: Criação Própria

3.6 INSTALAÇÃO E CONFIGURAÇÃO DO DNS (BIND)

A instalação do Servidor DNS é necessária, pois é nele que será realizada as consultas de nomes para a internet. A ferramenta instalada foi o BIND9. Sua versão é a 9.5.1.dfsg.P3-1+lenny1 para a implementação deste trabalho.

A Figura 76 mostra o comando para instalar o Servidor DNS:

```
# aptitude install bind9
```

Figura 76: Comando que instala o Servidor DNS

Fonte: Criação Própria

Após a instalação, deve-se entrar no diretório */etc/bind* e adicionar uma zona no arquivo *named.conf.local*. A Figura 77 mostra um exemplo de configuração de uma zona:

```
zone wifi.trt21.jus.br {
    type master;
    file "/etc/bind/wifi.db";
    notify no;
};
```

Figura 77: Exemplo de configuração de zona

Fonte: Criação Própria

Após criado a zona, criar um arquivo chamado *wifi.db* que especificará a configuração do domínio. A Figura 78 mostra um exemplo de configuração de domínio:

```

$TTL 604800
@ IN SOA ns.wifi.trt21.jus.br. root.wifi.trt21.jus.br. (
    2010031001 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@ IN NS ns.wifi.trt21.jus.br.
ns IN A 192.168.2.1
wpad IN A 192.168.2.1
proxy IN A 192.168.2.1
portal IN A 192.168.2.1

```

Figura 78: Exemplo de configuração de domínio DNS

Fonte: Criação Própria

Após criado o domínio, deve-se configurar o arquivo *named.conf.options* que especificará as consultas permitidas e os redirecionadores do provedor da internet. A Figura 79 mostra um exemplo de configuração:

```

acl rede-wpub { 192.168.2.0/24; };
acl loopback { 127.0.0.0/8; };
options {
    directory "/var/cache/bind";
    // IP da interface onde o named "escutara" as conexões
    listen-on { 192.168.2.1; 127.0.0.1; };

    // A principio apenas permite consultas da rede interna
    // isto poupa recursos e é mais seguro. Liberamos cada zona
    // individualmente...
    allow-query { rede-wpub; loopback; };

    // Permite consultas recursivas apenas a partir das redes indicadas
    allow-recursion { rede-wpub; loopback; };
    recursive-clients 2000;

    // DNS do Provedor
    forwarders {
        189.45.64.10; 189.45.112.2;
    };

    auth-nxdomain no; # conform to RFC1035
    listen-on-v6 { any; };
};

```

Figura 79: Exemplo de configuração do arquivo named.conf.options

Fonte: Criação Própria

Com a configuração finalizada do Servidor DNS, deve-se configurar o arquivo *resolv.conf* que fica do diretório */etc*, para que as requisições de nomes passem primeiro pelo DNS local e depois para o DNS Público. A Figura 80 mostra a configuração do arquivo *resolv.conf*

```
domain wifi.trt21.jus.br
search wifi.trt21.jus.br
nameserver 189.45.64.10
nameserver 189.45.112.2
```

Figura 80: Exemplo de configuração do arquivo resolv.conf

Fonte: Criação Própria

Reiniciar o Bind como mostra a Figura 81:

```
# /etc/init.d/bind9 restart
```

Figura 81: Comando que reinicia o Servidor DNS

Fonte: Criação Própria

3.7 INSTALAÇÃO E CONFIGURAÇÃO DO PROXY (SQUID)

A instalação do Servidor Proxy é necessária, pois é nele que será configurados os filtros de acesso a sites da internet e também fará cache dos sites para melhor desempenho de navegação da internet. Sua versão é a 2.7.STABLE3 para a implementação deste trabalho.

A Figura 82 mostra o comando para instalar o Servidor Proxy:

```
# aptitude install squid
```

Figura 82: Comando que instala o Servidor Proxy

Fonte: Criação Própria

Para configurar é necessário entrar na pasta */etc/squid/* e editar o arquivo *squid.conf*. A Figura 83 mostra algumas configurações adicionadas para controle de acesso e entre outras.

```

# Arquivo Squid.conf
#
http_port 192.168.2.1:8080 transparent

# IPs validos
# Rede Wifi TRT
acl IPvalido src 192.168.2.0/24
acl IPvalido src 127.0.0.1/32

# Controle de Acesso - ACL:
# -----
# Dominios explicitamente liberados
acl dominios_ok dstdomain "/etc/squid/dominios_ok"

# Dominios com download liberado (sem limite de tamanho)
acl download_ok dstdomain "/etc/squid/download_ok"

# Dominios explicitamente proibidos
acl dominios_bloq dstdomain "/etc/squid/dominios_bloq"
# Lista de palavras proibidas
acl palavras_bloq url_regex -i "/etc/squid/palavras_bloq"

# Arquivo onde serao guardados os logs de acesso do Squid.
cache_access_log /var/log/squid/access.log
# Permissoes de Acesso:
# -----
# Nega acesso a IPs nao permitidos
http_access deny !IPvalido

# Permite dominios liberados
http_access allow dominios_ok

# Bloqueia dominios proibidos
http_access deny dominios_bloq

# Bloqueia urls q contenham palavras improprias
http_access deny palavras_bloq

# Permite IPs validos
http_access allow IPvalido

# Libera geral para os micros liberados e dominios especificos
reply_body_max_size 0 allow all download_ok

# Permite ate 150 MB para os dominios liberados
reply_body_max_size 157286400 allow all dominios_ok

# Limita a 50MB (50*1024*1024) para os demais dominios
reply_body_max_size 52428800 allow all

```

Figura 83: Exemplo de configuração adicionadas no arquivo squid.conf

Fonte: Criação Própria

Como mostra na Figura 83, existem ACL's (Listas de Controle de Acesso) para fazer o controle de acesso a internet. Para configurar o conteúdo dessas listas, as Figuras 85 a 87 mostram exemplos de configurações:

```
.xvideos.com
.kortaz.com
.hidemyass.com
```

Figura 84: Exemplo de configuração do arquivo dominios_bloq. ACL para domínios bloqueados

Fonte: Criação Própria

```
.broffice.org
.sun.com
```

Figura 85: Exemplo de configuração do arquivo dominios_ok. ACL para domínios permitidos

Fonte: Criação Própria

```
ubuntu.c3sl.ufpr.br
.oracle.com
.redhat.com
```

Figura 86: Exemplo de configuração do arquivo downloads_ok. ACL para downloads permitidos

Fonte: Criação Própria

```
kaaza
michupa
xxx
facegoo
mikeinbrazil
safada
```

Figura 87: Exemplo de configuração do arquivo palavras_bloq. ACL para palavras bloqueadas

Fonte: Criação Própria

Após configuração, é necessário reiniciar o Squid como mostra a Figura 88:

```
# /etc/init.d/squid restart
```

Figura 88: Comando que reinicia o Servidor Proxy

Fonte: Criação Própria

Para que não seja utilizado o Proxy transparente, pois ele tem o problema de não acessar páginas seguras (https), é necessário utilizar o protocolo WPAD (*Web Proxy Autodiscovery Protocol*), ele será responsável pela detecção automática do Proxy nos browsers. Para seu funcionamento, será necessário criar um arquivo chamado *wpad.dat*, na pasta */var/www/proxy/*.

A Figura 89 mostra um exemplo de configuração:

```
function FindProxyForURL(url, host){
  if (isPlainHostName(host) ||
      isInNet(host,"192.168.2.0","255.255.255.0") ||
      dnsDomainIs(host,".wifi.trt21.jus.br"))
    return "DIRECT";
  else
    return "PROXY 192.168.2.1:8080";
}
```

Figura 89: Exemplo de configuração do arquivo wpad.dat

Fonte: Criação Própria

Criado o arquivo será necessário configura o DHCP, o DNS, o Proxy e o Apache, esses mostrarão ao host a configuração do Proxy.

No DHCP será incluído algumas configurações como mostra a Figura 90.

```
option wpad code 252 = text;
option wpad "http://192.168.2.1/wpad.dat\n";
```

Figura 90: Exemplo de configuração adicionada ao DHCP para o WPAD

Fonte: Criação Própria

No DNS será incluído uma configuração como mostra a Figura 91.

wpad	IN	A	192.168.2.1
------	----	---	-------------

Figura 91: Exemplo de configuração adicionada ao DNS para o WPAD

Fonte: Criação Própria

No Proxy será incluído uma configuração como mostra a Figura 92.

```
header_access Via deny all
header_access X-Forwarded-For deny all
header_access Proxy-Connection deny all
header_access Accept-Encoding deny all
```

Figura 92: Exemplo de configuração adicionada ao Proxy para o WPAD

Fonte: Criação Própria

No Apache será criado um Virtualhost como mostra a Figura 70 exibida anteriormente.

Agora, configura-se o cliente para detectar de forma automática o Proxy para acessar a internet, conforme a Figura 93:

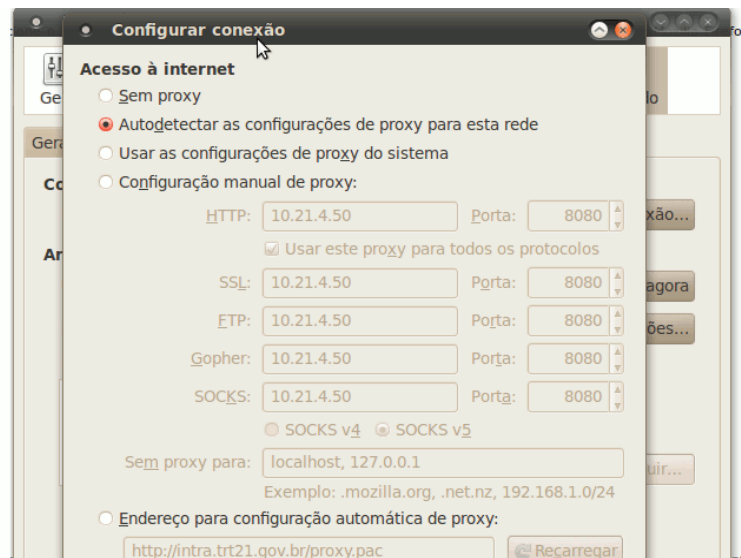


Figura 93: Exemplo de configuração do proxy no cliente

Fonte: Criação Própria

Com o Proxy configurado, será necessário efetuar uns ajustes no Nocat, no diretório `/usr/local/nocat/gw/bin` onde fica o arquivo `initialize.fw`.

A Figura 94 mostra o código padrão onde será feita a alteração:

```
for port in 80 443; do
  for mark in $nonauth; do
    $redirect -m mark --mark $mark -d windowsupdate.microsoft.com -j RETURN
    $redirect -m mark --mark $mark -p tcp --dport $port -j REDIRECT \
      --to-port $GatewayPort
  done
done
```

Figura 94: Exemplo de configuração do código padrão do arquivo `initialize.fw`

Fonte: Criação Própria

A Figura 95 mostra o código após a alteração:

```
for port in 80 443 8080; do
  for mark in $nonauth; do
    $redirect -m mark --mark $mark -d windowsupdate.microsoft.com -j RETURN
    $redirect -m mark --mark $mark -p tcp --dport $port -j REDIRECT \
      --to-port $GatewayPort
  done
done
```

Figura 95: Exemplo de configuração do código após alteração do arquivo `initialize.fw`

Fonte: Criação Própria

Terminada a configuração, é necessário reiniciar o Nocat. A Figura 96 mostra o comando para reiniciar:

```
# /etc/init.d/nocat restart
```

Figura 96: Comando que reinicia o Nocat

Fonte: Criação Própria

Caso aconteça erro de redirecionamento no browser após a autenticação no Nocat, como mostra a Figura 97:

```
http://www.google.comwww.google.com
```

Figura 97: Exemplo de erro após autenticação no Nocat

Fonte: Criação Própria

Terá que ser feito um ajuste no arquivo *Gateway.pm* que fica no diretório */usr/local/nocat/gw/lib/NoCat*.

A Figura 98 mostra o código padrão que será feito a alteração:

```
$head{URL} = ($head{Host} ? "http://$head{Host}$head{URI}" : $self->{HomePage}) || "";
```

Figura 98: Exemplo do código padrão do arquivo gateway.pm

Fonte: Criação Própria

A Figura 99 mostra o código após a alteração:

```
$head{URL} = ($head{Host} ? "$head{URI}" : $self->{HomePage}) || "";
```

Figura 99: Exemplo do código após alteração do arquivo gateway.pm

Fonte: Criação Própria

Terminada a configuração, é necessário reiniciar o Nocat. A Figura 96 mostra o comando para reiniciar.

3.8 INSTALAÇÃO E CONFIGURAÇÃO DO NTP

A instalação do NTP é necessária, pois é ele que fará a sincronização de data e hora através de fontes confiáveis disponíveis na internet, mantendo assim o Sistema Operacional sempre com a data e hora atualizada. Sua versão é a 1:4.2.4p4+dfsg-8lenny3 para a implementação deste trabalho.

A Figura 100 mostra o comando para instalar o NTP:

```
# aptitude install ntp ntpdate
```

Figura 100: Comando que instala o NTP

Fonte: Criação Própria

Para configurar é necessário entrar na pasta `/etc/` e editar o arquivo `ntp.conf`. A Figura 101 mostra as configurações para ser adicionadas:

```
# Servidores Publicos
server ntp.pop-rn.rnp.br
server ntp.cais.rnp.br

# Restrições de acesso
restrict 192.168.2.0 mask 255.255.255.0 nomodify
```

Figura 101: Exemplo de configuração de servidores públicos no arquivo ntp.conf

Fonte: Criação Própria

Terminada a configuração, é necessário reiniciar o NTP. A Figura 102 mostra o comando para reiniciar.

```
# /etc/init.d/ntp restart
```

Figura 102: Comando que reinicia o NTP

Fonte: Criação Própria

3.9 INSTALAÇÃO E CONFIGURAÇÃO DO SARG

A instalação do SARG é necessária, pois é ele que fará a geração de relatórios de acesso a internet dos usuários através dos logs do Proxy (SQUID). Sua versão é a 2.2.5-2 para a implementação deste trabalho.

A Figura 103 mostra o comando para instalar o SARG:

```
# aptitude install sarg
```

Figura 103: Comando que instala o SARG

Fonte: Criação Própria

Para configurar é necessário entrar na pasta `/etc/squid/` e editar o arquivo `sarg.conf`. A Figura 104 mostra as configurações a serem realizadas:

```
language Portuguese
topsites_num 50
topsites_sort_order BYTES D
```

Figura 104: Exemplo de configuração a ser realizada no arquivo sarg.conf

Fonte: Criação Própria

Quando o SARG é instalado, é criado uns scripts para execução no Cron. Mas esses scripts serão removidos. A Figura 105 mostra os comandos para a remoção dos scripts:

```
# rm /etc/cron.daily/sarg
# rm /etc/cron.monthly/sarg
# rm /etc/cron.weekly/sarg
```

Figura 105: Comandos que fazem a remoção dos scripts do Sarg

Fonte: Criação Própria

Após a remoção dos scripts, será necessário criar um novo script para a criação dos relatórios de acesso. A Figura 106 mostra um exemplo do script chamado *sarg-dia.sh*.

```
#!/bin/bash
/usr/bin/sarg -f /etc/squid/sarg.conf
exit 0
```

Figura 106: Exemplo do script sarg-dia.sh

Fonte: Criação Própria

Depois de criado, dar permissão de execução ao arquivo como mostra a Figura 107:

```
# chmod +x sarg-dia.sh
```

Figura 107: Comando que dar permissão de execução ao script sarg-dia.sh

Fonte: Criação Própria

E depois adicioná-lo no arquivo */etc/crontab*, que será colocado para ser executado todos os dias às 23:50. A Figura 108 mostra a configuração adicionada:

```
50 23 * * * root /etc/squid/sarg-dia.sh
```

Figura 108: Exemplo de configuração adicionada ao crontab para execução do script sarg-dia.sh

Fonte: Criação Própria

Para a visualização dos relatórios gerados pelo SARG, deve-se acessar a partir de um browser, acessando o endereço <http://ipdoservidor/squid-reports>, conforme a Figura 109:

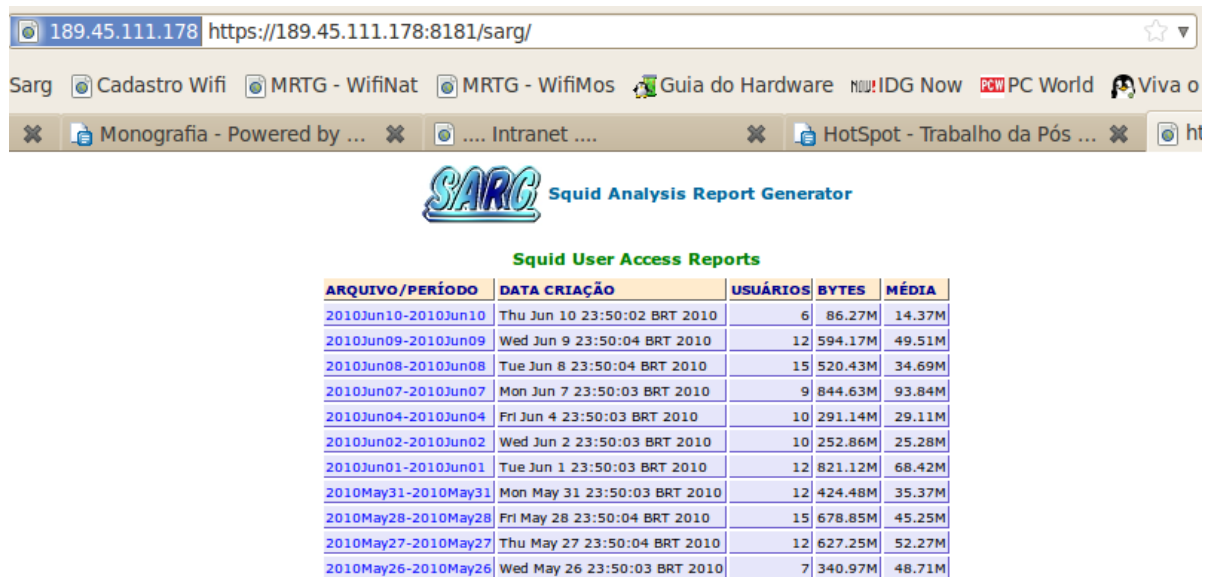


Figura 109: Relatório gerado pelo Sarg

Fonte: Criação Própria

3.10 INSTALAÇÃO E CONFIGURAÇÃO DO MRTG COM SNMP

A instalação do MRTG é necessária, pois é ele que fará a geração dos gráficos com os dados recolhidos, para a monitoração do link da internet. Sua versão é a 2.16.2-3 para a implementação deste trabalho.

A Figura 110 mostra o comando para instalar o MRTG e o SNMP:

```
# aptitude install snmp snmpd mrtg
```

Figura 110: Comando que instala o MRTG e o SNMP

Fonte: Criação Própria

Primeiro será configurado o SNMP. Para configurá-lo deve-se entrar na pasta `/etc/snmp/` e editar o arquivo `snmpd.conf`.

A Figura 111 mostra a configuração a ser realizada:

```
#   sec.name source      community
#com2sec paranoid default public
com2sec readonly default public
#com2sec readwrite default private

syslocation TRT21
syscontact sysadm@trt21.jus.br
```

Figura 111: Exemplo de configuração da comunidade habilitada no arquivo snmpd.conf

Fonte: Criação Própria

Após configuração, deve-se reiniciar o SNMP como mostra a Figura 112:

```
# /etc/init.d/snmpd restart
```

Figura 112: Comando que reinicia o SNMP

Fonte: Criação Própria

Para verificar se o SNMP está configurado corretamente, usa-se o comando *snmpwalk*. A Figura 113 mostra o comando que faz a verificação:

```
# snmpwalk -v 2c -c public localhost
```

Figura 113: Comando que verifica a conexão com o SNMP

Fonte: Criação Própria

Terminada a configuração do SNMP, será configurado o MRTG. Para isso será criado um arquivo chamado *mrtg.cfg*, aonde nele será armazenado as configurações das interfaces de rede. A Figura 114 mostra o comando que gera as configurações:

```
# cfgmaker public@localhost > /etc/mrtg.cfg
```

Figura 114: Comando que gera as configurações do MRTG

Fonte: Criação Própria

Gerado o arquivo *mrtg.cfg*, deve-se configurar a opção *MaxBytes*, que é a velocidade do link do Provedor de Internet. A Figura 115 mostra um exemplo de um link de 2 Megas:

```
MaxBytes[localhost_3]: 256000
```

Figura 115: Exemplo de configuração de um link de 2 Megas.

Fonte: Criação Própria

Terminada a configuração, será criado uma página de acesso para visualização dos gráficos. A Figura 116 mostra o comando que cria a página:

```
# indexmaker /etc/mrtg.cfg > /var/www/mrtg/index.html
```

Figura 116: Comando que cria a página de acesso aos gráficos do MRTG

Fonte: Criação Própria

Para a visualização dos gráficos gerados pelo MRTG, deve-se acessar a partir de um browser, acessando o endereço <http://ipdoservidor/mrtg>, conforme a Figura 117:

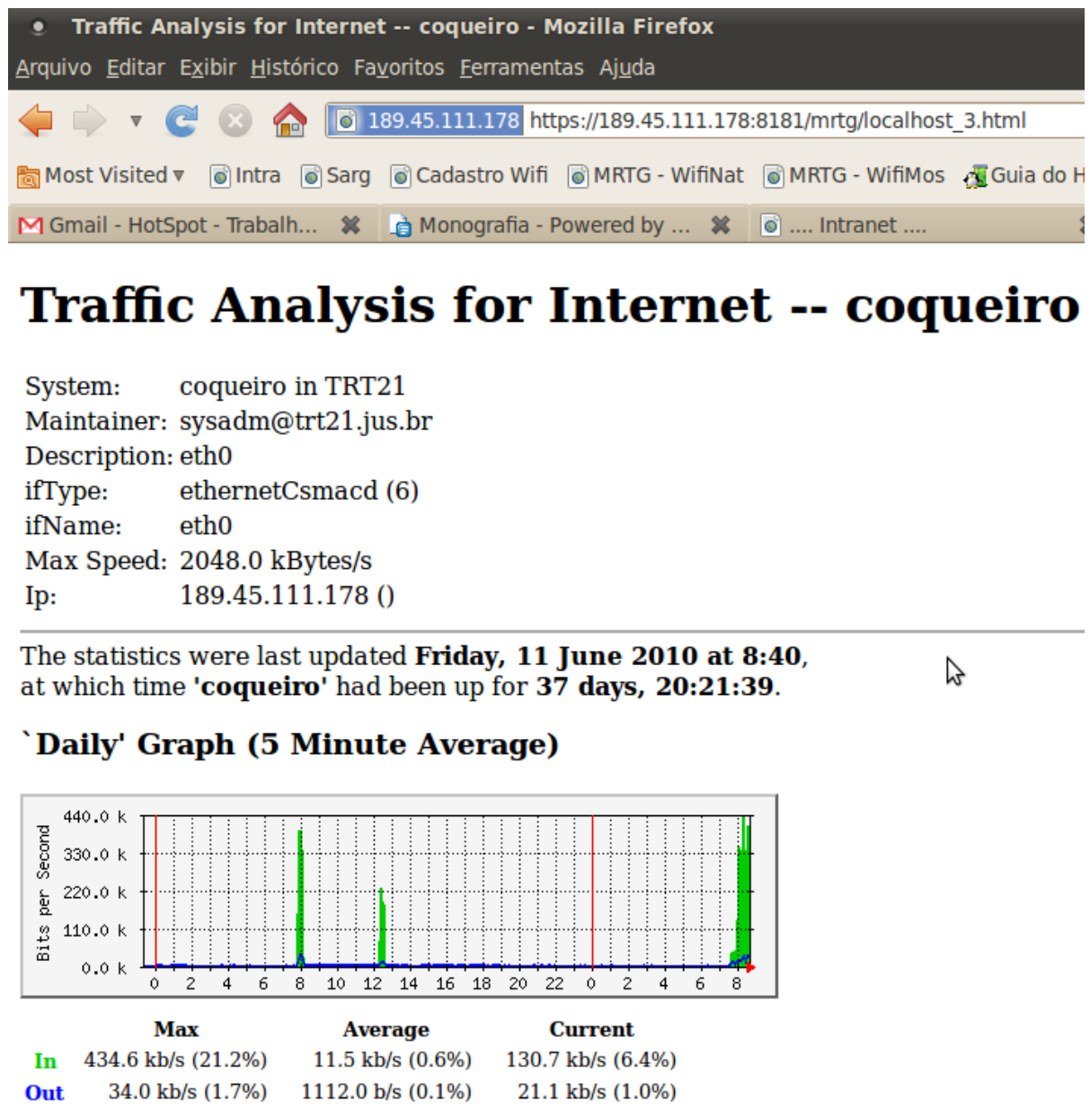


Figura 117: Gráficos gerados pelo MRTG

Fonte: Criação Própria

3.11 PORTAL DE CADASTRO DE USUÁRIOS

Inicialmente os cadastros dos usuários eram realizados por scripts, mas para ter acesso aos scripts era necessário acessar o servidor por conexão SSH. Isso era um problema pois nem todos poderiam ter acesso direto ao servidor e nem todos saberiam executar tal função. Então foi pensado a criação de um Portal que fizesse tudo de forma mais transparente possível, podendo qualquer pessoa efetuar o cadastro dos usuários, desde que essa pessoa possuía um cadastro realizado previamente pelo administrador do Portal. O Portal foi desenvolvido na linguagem PHP.

Para ter acesso ao Portal, deve-se acessar a partir de um browser, acessando o endereço `https://ipdoservidor:8181/wifitr/`, onde será exibida a tela de autenticação conforme a Figura 118:



Figura 118: Tela inicial do Portal de Cadastro

Fonte: Criação Própria

Depois de autenticado, será exibida a tela de Cadastro de Usuários, onde com o preenchimento dos dados, os usuários poderão ter acesso a Internet sem fio. A Figura 119 mostra a tela de cadastro de usuários:

Figura 119: Tela de Cadastro dos usuários

Fonte: Criação Própria

Para cadastro de novos administradores do Portal, deve-se acessar o menu Cadastros Administradores. A Figura 120 mostra a tela de cadastro de administradores:



- Cadastros
 - Cientes
 - Administradores
- Consultas
 - Cientes
- Geral
 - Alterar Senha - Cliente Wireless
 - Alterar Senha - Administrador
 - Logout

[Cadastro de Administradores](#)

Login*

Senha*

Repita a Senha*

☐ Limitado

Figura 120: Tela de Cadastro de administradores

Fonte: Criação Própria

Para efetuar consultas de usuários, deve-se acessar o menu Consultas Clientes e fornecer o CPF do usuário. A Figura 121 mostra a tela de consulta de usuários:



- Cadastros
 - Cientes
 - Administradores
- Consultas
 - Cientes
- Geral
 - Alterar Senha - Cliente Wireless
 - Alterar Senha - Administrador
 - Logout

[Consulta de Clientes](#)

Informe o CPF

Ex: 12345678900
ou a palavra: todos

Host	Nome	Login	CPF	Cadastrado por
	Thales Eduardo Bernardo de Figueiredo	thales.3lan	0987654321	thales.3lan

Figura 121: Tela de Consulta de usuários

Fonte: Criação Própria

Para consultar, cadastrar ou excluir os equipamentos dos usuários, após a consulta do usuário, deve-se clicar na lupa da coluna Host. Com isso será exibido se o usuário possui ou não equipamento cadastrado, conforme a Figura 122.

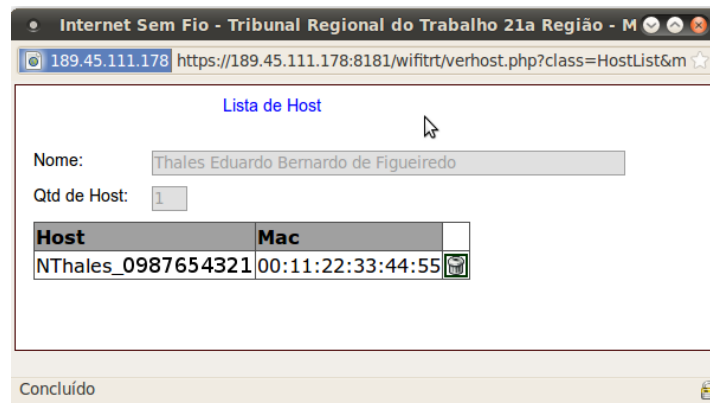


Figura 122: Tela de Consulta e/ou Cadastrado de equipamento dos usuários

Fonte: Criação Própria

4 CONCLUSÃO

Este trabalho mostrou uma de várias outras formas de como montar um Hotspot e como integrar vários tipos de ferramentas para sua implantação, utilizando as vantagens de cada ferramenta para exercer seu papel escolhido.

As maiores dificuldades encontradas na implementação do Hotspot foi fazer a integração das ferramentas Nocat, Radius, Ldap, Squid fazendo com que cada uma trabalhasse em conjunto uma com outra.

Como dito anteriormente, existem várias formas de montar um Hotspot, caso desejem fazer novos trabalhos, segue algumas sugestões com novas ferramentas que também fazer o papel de Captive Portal, como por exemplo o Wifidog, PFSense, Mikrotik.

Um ponto importante é que todas as ferramentas são *open source* (código aberto). Isso mostra que apesar das ferramentas pagas existentes no mercado, podemos utilizar ferramentas livres que fazem o mesmo papel.

REFERÊNCIAS

BRANDÃO, Patrick. 2004, **NoCatAuth - Construindo um firewall/gateway autenticado**. Disponível em: <<http://www.vivaolinux.com.br/artigo/NoCatAuth-Construindo-um-firewall-gateway-autenticado/>> Acesso em 10 abr. 2010.

FREIRE, Vinícius Pires de Moura. 2007, **Sistema de Segurança sem Fio para o POP-PI/RNP**. Disponível em: <<http://www.pop-pi.rnp.br/artigos/Sistema%20de%20Seguranca%20em%20Redes%20Wireless.pdf>> Acesso em 30 abr. 2010.

GEYER Cláudio, 2010, **Manual OpenLDAP**. Disponível em: <<http://www.inf.ufrgs.br/gppd/disc/inf01008/trabalhos/sem01-1/t1/openldap/>> Acesso em 20 maio. 2010.

KASHIWAKURA, Milton Kaoru. 2010, **Instalação e Configuração do NTP**. Disponível em: <<http://ntp.br/utilizando.html>> Acesso em 03 maio. 2010.

LACERDA, Gleidson. 2009, **Apache 2 com SSL (HTTPS)**. Disponível em: <<http://wiki.ubuntu-br.org/Apache2%2BSSL>> Acesso em 30 abr. 2010.

OETIKER, Tobias. 2009, **The MRTG 2.16.2 Linux/Unix Installation Guide**. Disponível em: <<http://oss.oetiker.ch/mrtg/doc/mrtg-unix-guide.en.html>> Acesso em 10 maio. 2010.

POLESSO, Rodrigo. 2004, **Gerenciando relatórios do SARG**. Disponível em: <<http://www.vivaolinux.com.br/artigo/Gerenciando-relatorios-do-SARG/>> Acesso em 01 maio. 2010.

ROSA, Marcos Antonio. 2007, **Servidor de autenticação 802.1x com Freeradius**. Disponível em: <<http://www.vivaolinux.com.br/artigo/Servidor-de-autenticacao-802.1x-com-Freeradius/?pagina=3>> Acesso em 11 abr. 2010.

SANTOS, Luciano. 2008, **Como instalar MRTG e SQUID no Ubuntu**. Disponível em: <<http://luciano.wordpress.com/2007/03/13/como-instalar-mrtg-e-squid-no-ubuntu/>> Acesso em 10 maio. 2010.