APOSTILA

ADMNISTRAÇÃO DE REDES LINUX

Índice

Conceitos de Redes e Protocolos

TCP/IP

Os protocolos TCP/IP são um conjunto de protocolos de comunicação que definem como tipos diferentes de computadores conversam uns com os outros. O seu nome vem dos dois protocolos mais comuns: o Protocolo de Controle de Transmissão (TCP – Transmission Control Protocol) e o Protocolo de Internet (IP – Internet Protocol). O Protocolo de Internet transmite dados na forma de datagramas entre computadores; divide os dados em pacotes, que são enviados para os computadores via rede. O Protocolo de Controle de Transmissão assegura que os datagramas em uma mensagem serão remontados na ordem correta para seu destino final, e que os datagramas que estão faltando serão reenviados até que sejam corretamente recebidos. Outros protocolos que fazem parte do TCP/IP são:

ARP	Address Resolution Protocol (Protocolo de Resolução de Endereço). Traduz endereços Internet para endereços locais de hardware.
ICMP	Internet Control Message Protocol (Protocolo Internet de Controle de Mensagens). É o protocolo de controle de mensagens.
PPP	Point to Point Protocol (Protocolo Ponto a Ponto). Proporciona conexões de rede síncronas e assíncronas.
RARP	Reverse Address Resolution Protocol (Protocolo de Resolução de Endereço Reverso). Traduz endereços locais de hardware para endereços Internet. O oposto do ARP.
SLIP	Serial Line Internet Protocol (Protocolo Internet de Linha Serial). Habilita IP em linhas seriais.
SMTP	Simple Mail Transport Protocol (Protocolo Simples de Transporte de Correio). Usado pelo sendmail e pelo postfix para enviar mensagens de correio eletrônico via TCP/IP.
SNMP	Simple Network Management Protocol (Protocolo Simples de Gerenciamento de Rede). Realiza funções distribuídas de gerenciamento de rede via TCP/IP.
UDP	User Datagram Protocol (Protocolo de Datagrama de Usuário).

Camadas do conjunto de protocolos TCP/IP

O conjunto de protocolos TCP/IP possui quatro camadas que serão descritas a seguir:

Camada de aplicação

Permite o desenvolvimento e a utilização de aplicações pelo usuário, possuindo vários protocolos como SMTP, TELNET, FTP, DNS, TFTP, RPC, NFS, SNMP, SSH, HTTP etc.

Camada de transporte

É a responsável por receber os dados enviados pela camada de aplicação e dividi-los em pacotes, que serão enviados para a Camada de Internet. É onde estão localizados os protocolos TCP, responsáveis pelo transporte seguro (entrega garantida de informação) de pacotes entre o nó de origem e o nó de destino, e o UDP, responsável pelo transporte inseguro (entrega não garantida de informação). Estabelece conexões virtuais em que aplicações não precisam gerenciar retransmissão, controle de seqüência, perda de integridade e controle de fluxo (TCP), ou disponibiliza o serviço de datagrama (UDP), em que as aplicações garantem o transporte seguro. Essa camada é incorporada pelo sistema operacional.

Camada de Internet

É a responsável por receber os dados enviados pela camada de transporte e dividi-los em datagramas que serão enviados para a Camada de rede. É onde estão localizados os protocolos IP (responsáveis pelo roteamento e retransmissão de pacotes para a rede, até a mensagem chegar ao destino), ICP, ARP e RARP. Essa camada é incorporada pelo sistema operacional.

Camada de rede

Esta camada é responsável por enviar os datagramas pela rede.

Aplicação
Transporte
Internet
Rede

Camadas dos protocolos TCP/IP

RFCs

Os detalhes de cada protocolo TCP/IP são descritos em documentos conhecidos como RFCs (Requests For Comments – Solicitações Para Comentários). Esses documentos são distribuídos gratuitamente pela Internet e continuam a evoluir conforme surgem novas tecnologias e técnicas.

Para maiores informações sobre as RFCs, veja o site www.rfc-editor.org.

Endereços IP

O endereço IP é um número binário de 32 bits (IPV4) atribuído a interfaces de rede em computadores (hosts), que diferencia um computador dos demais na rede. Na maioria dos casos, os computadores que estão ligados na rede têm somente uma interface de rede, possuindo como resultado apenas um endereço IP. Porém, os computadores e outros dispositivos que estão ligados na rede podem ter várias interfaces de rede, e cada uma delas possui o seu próprio endereço IP. Dessa forma, um dispositivo com quatro interfaces de rede (um roteador, por exemplo) terá quatro endereços IP, um endereço para cada interface de rede.

Endereços IP de 32 bits (IPV4)

Na implementação atual de endereços IP (IPV4), os endereços consistem em 4 bytes (lembre-se de que 1 byte contém 8 bits), resultando em um total de 32 bits. A convenção é escrever cada byte como um número decimal e colocar um ponto (.) após cada número; por exemplo, 145.97.23.106. Essa forma de escrever endereços IP é conhecida como notação de ponto decimal.

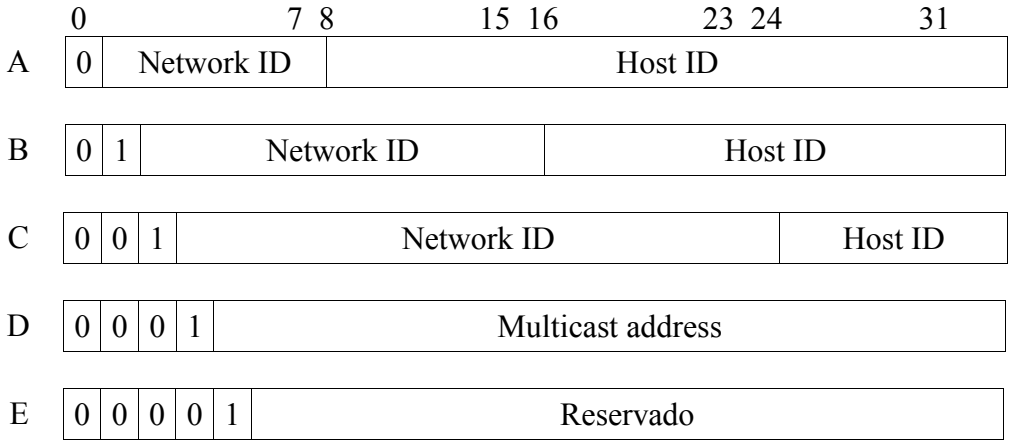
Classes de endereços IP

Os bits de um endereço IP são interpretados da seguinte forma:

<endereço_rede> <endereço_host>

Em outras palavras, um número determinado de bits do endereço IP de 32 bits é reservado a um endereço de rede e o restante dos bits é reservado para um endereço de host. O endereço de rede identifica a rede à qual o computador está ligado e o endereço de host identifica o computador.

Para acomodar redes de vários tamanhos (o tamanho da rede é o número de computadores naquela rede), o endereço IP inclui o conceito de diversas classes de rede. Existem cinco classes de endereço IP, conforme mostrado na figura abaixo:



Classes de endereços IP.

Das cinco classes de rede, apenas as classes A, B e C são utilizadas para endereçamento de redes e hosts; as classes D e E são reservadas a uso especial.

As classes de endereços IP podem ser reconhecidas pelo primeiro número na notação em ponto decimal, como a seguir:

- Endereços da classe A: 1.xxx.xxx.xxx a 126.xxx.xxx.xxx:
  Número de redes permitido = 1.0.0.0 a 126.0.0.0 = 126
  Número máximo de hosts em cada rede =[(2^24)-2]=16.777.214

- Endereços da classe B: 128.xxx.xxx.xxx a 191.xxx.xxx.xxx:
  Número de redes permitido = 128.0.0.0 a 191.255.0.0 = 2^14=16384
  Número máximo de hosts em cada rede =[(2^16)-2]=65.534

- Endereços da classe C: 192.xxx.xxx.xxx a 223.xxx.xxx.xxx:

Número de redes permitido = 192.0.0.0 a 223.255.254.0 = 2^21=20.971.152
Número máximo de hosts em cada rede =[(2^8)-2]=254

- Endereços da classe D: 224.xxx.xxx.xxx a 239.xxx.xxx.xxx:

  O endereço de multicast é utilizado na transmissão simultânea de um ou mais pacotes para um grupo de hosts, sendo identificados por um endereço especial de destino (multicast address).

- Endereços de classe E: 240.xxx.xxx.xxx a 255.xxx.xxx.xxx:
  Reservado para uso futuro pelo InterNIC.

Endereços IP para a rede

Em uma rede TCP/IP, todos os computadores da rede têm um endereço IP e um nome. Se uma rede não está conectada à Internet, ela não precisa de endereços IP exclusivos. A RFC 1918 ("Alocação de Endereços para Redes Internet Privadas") fornece diretrizes sobre quais endereços IP podem ser utilizados dentro de redes Internet privadas (o termo Internet privada é o mesmo que Intranet). Esses endereços não são roteados para a Internet. Três faixas de endereços IP estão reservadas para redes Internet privadas:

Endereços da classe	Faixa		
A	10.0.0.0	a	10.255.255.255
B	172.16.0.0	a	172.31.255.255
C	192.168.0.0	a	192.168.255.255

Uma rede que utiliza endereços IP reservados para Internet privada pode ter acesso à Internet utilizando NAT (Network Address Translator – Tradutor de endereço de rede).

Endereço de loopback

O endereço 127.xxx.xxx.xxx da classe A é utilizado para loopback (comunicações dentro do mesmo host). Convencionalmente, 127.0.0.1 é utilizado como endereço de loopback. Processos que precisam comunicar-se por meio de TCP com outros processos no mesmo host utilizam o endereço loopback para não ter que enviar pacotes na rede.

Endereço de rede

O endereço de uma rede sempre terá todos os bits do endereço de host configurados para "0", a não ser que a rede seja dividida em sub-redes. Na realidade, resulta de um "AND lógico" entre os bits do endereço IP de um host e os bits da máscara de rede desse mesmo host.

Endereço de broadcast

O endereço de broadcast de uma rede sempre terá todos os bits do endereço de host configurados para "1", a não ser que a rede seja dividida em sub-redes.

Endereços IP de 128 bits (IPV6)

No momento em que o endereço IP de 4 bytes foi criado, o número de endereços parecia adequado. Entretanto, os endereços IP estão se esgotando e a IETF (Internet Engineering Task Force) reconheceu

este problema em 1991 e começou a trabalhar no esquema de endereçamento IP da próxima geração, chamado IPV6, que eventualmente substituirá o esquema de endereçamento atual (IPV4). O esquema de endereçamento IPV6 consiste em endereços IP de 16 bytes, resultando em um total de 128 bits. O Linux tem suporte ao IPV6.

Máscara de rede

A máscara de rede não é um endereço IP. Ela serve para identificar a classe de rede e se esta está dividida em sub-redes ou não. A máscara de rede tem todos os bits do endereço da rede configurados para "1" e todos os bits de endereço do host configurados para "0",a não ser que a rede seja dividida em sub-redes.

Arquitetura cliente-servidor

Um serviço de Internet típico é implementado em duas partes: a primeira é implementada em um computador servidor, que fornece informações, e a segunda é implementada em um ou mais computadores clientes, que solicitam informações. Tal arquitetura é conhecida como arquitetura cliente/servidor.

Serviços, portas e soquetes do TCP/IP

Um serviço pode ser definido como uma aplicação que necessita de um protocolo e opera em uma porta. O arquivo /etc/services contém a lista de serviços disponíveis para o Linux/Unix e suas respectivas portas e protocolos utilizados.

Exemplo de uma parte do arquivo /etc/services:

```
#vi /etc/services

# /etc/services:
# $Id: services,v 1.4 1997/05/20 19:41:21 tobias Exp $
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1700, ``Assigned Numbers'' (October 1994).  Not all ports
# are included, only the more common ones.

tcpmux          1/tcp                   # TCP port service multiplexer
tcpmux          1/udp
compressnet     2/tcp                   # Management Utility
compressnet     2/udp
compressnet     3/tcp                   # Compression Process
compressnet     3/udp
```

```
#
#
rje             5/tcp                   # Remote Job Entry
rje             5/udp
#
#
echo            7/tcp                   # Echo
echo            7/udp
#
#
discard         9/tcp           sink null       # Discard
discard         9/udp           sink null
#
#
systat          11/tcp          users           # Active Users
systat          11/udp      users
#
#
daytime         13/tcp                  # Daytime
daytime         13/udp
#
#
netstat         15/tcp
#
#
qotd            17/tcp          quote           # Quote of the Day
qotd            17/udp      quote
msp             18/tcp                      # message send protocol
msp             18/udp
chargen         19/tcp          ttytst source    # Character Generator
chargen         19/udp          ttytst source
ftp-data        20/tcp                  # File Transfer [Default Data]
ftp-data        20/udp
ftp             21/udp                  # File Transfer [Control]
ftp             21/tcp
#fsp            21/udp          fspd
ssh             22/tcp                      # SSH Remote Login Protocol
ssh             22/udp
telnet          23/tcp                  # Telnet
telnet          23/udp
#
#
smtp            25/tcp          mail            # Simple Mail Transfer
smtp            25/udp      mail
#
#
nsw-fe          27/tcp                  # NSW User System FE
```

```
nsw-fe          27/udp
#
#
msg-icp         29/tcp                  # MSG ICP
msg-icp         29/udp
#
#
msg-auth        31/tcp                  # MSG Authentication
msg-auth        31/udp
#
```

Uma porta pode ser definida como um canal de comunicação para um computador. Pacotes de dados que chegam a um computador não são apenas endereçados ao computador, e sim a este computador em uma determinada porta. As aplicações-padrão utilizam sempre uma mesma porta. Por exemplo: o protocolo HTTP utiliza sempre a porta 80 e o protocolo SMTP, a porta 25. A utilização de um número de porta permite ao protocolo TCP da camada de transporte saber qual é o tipo de conteúdo do pacote de dados e, no receptor, saber para qual protocolo de aplicação deverá entregar esse pacote de dados. Dessa forma, ao receber um pacote de dados endereçado à porta 25, o protocolo TCP irá entregá-lo ao protocolo que estiver operando nessa porta, o SMTP, que por sua vez, entregará o pacote de dados à aplicação que o solicitou (programa de e-mail). O número de portas TCP/IP disponível é 65536.

Um soquete pode ser definido como uma conexão dentro de uma porta. Dessa forma é possível ter várias conexões diferentes abertas em uma mesma porta. Por exemplo: dois browsers podem estar abertos simultaneamente e cada um deles carregado com um site de Internet diferente, ou seja, eles utilizam a mesma porta, mas soquetes diferentes. A camada de aplicação saberá para qual browser deverá entregar os dados recebidos, em virtude de que com os dados recebidos vem a informação de qual é o soquete de destino que foi inserido pela camada de aplicação do transmissor. Existem dois tipos de soquete: soquete de fluxo, utilizado para troca de dados em protocolos orientados por conexão, e soquete de datagrama, utilizado para troca de dados em protocolos sem conexão.

TCP

Garante a entrega dos dados transmitidos e requer o estabelecimento de uma conexão entre o transmissor e o receptor. Seu funcionamento é análogo a uma conversa telefônica comum. Quando queremos falar com um amigo, precisamos primeiro discar o seu número de telefone e estabelecer uma conexão antes de ser possível conversar. A troca de dados orientada por conexão requer que tanto o processo de envio como o processo de recepção estabeleçam uma conexão antes que a troca de dados possa começar. O protocolo TCP (Transmission Control Protocol – Protocolo de Controle de Conexão) é um exemplo de protocolo orientado por conexão.

UDP

Não garante a entrega dos dados transmitidos e não requer o estabelecimento de uma conexão entre o transmissor e o receptor. É como gritar para um amigo em uma sala cheia; nunca poderemos ter certeza se de fato ele escutará. O protocolo UDP (User Datagram Protocol – Protocolo de Datagrama do Usuário) é um exemplo de protocolo sem conexão.

Configurando uma estação na rede

Xinetd

O xinetd é o superdaemon que combina as funções dos daemons inetd e tcpd. O xinetd tem um arquivo de configuração chamado /etc/xinetd.conf e um diretório chamado /etc/xinetd.d, que contém um arquivo de configuração para cada serviço. Isso permite a um pacote como o wu-ftpd controlar a sua própria configuração por meio de um arquivo separado.

Exemplo do arquivo /etc/xinetd.conf:

```
defaults
{
      instances            = 25
      per_source           = 10
      log_type             = SYSLOG authpriv
      log_on_success  = HOST PID USERID
      log_on_failure = HOST USERID
}

includedir /etc/xinetd.d
```

Exemplo do arquivo /etc/xinetd.d/telnet:

```
service telnet
{
      disable        = no
      flags          = REUSE
      socket_type    = stream
      protocol       = tcp
      wait           = no
      user           = root
      server         = /usr/sbin/in.telnetd
      log_on_failure += USERID
}
```

O significado desses parâmetros é dado a seguir:

instances -> Define o número máximo de sessões de servidor. No arquivo /etc/xinetd.conf exemplo, o número máximo de usuários conectados ao servidor FTP é 25.

per_source -> Define o número máximo de conexões simultâneas a um servidor a partir da origem (endereço). No arquivo /etc/xinetd.conf exemplo, apesar de podermos ter 25 pessoas conectadas ao servidor FTP ao mesmo tempo, só podemos ter 10 conexões vindas de um mesmo endereço.

log_type -> Define qual a maneira que o xinetd utilizará para fazer um registro de acesso. No arquivo /etc/xinetd.conf exemplo, foi utilizado o syslog na seção authpriv.

log_on_success -> Define o tipo de informações que devem ser registradas em caso de sucesso na conexão.

log_on_failure -> Define o tipo de informações que devem ser registradas em caso de falha na conexão.

includedir -> Define o diretório onde são armazenados os arquivos de configuração dos servidores.

disable -> Define se o servidor está desabilitado (yes) ou habilitado (no).

flags -> Configura opções especiais para a execução do servidor. No arquivo / etc/xinetd.d/ftp exemplo, permite que o daemon tcpd seja utilizado como servidor e o in.ftpd -l -a como argumento deste, do mesmo modo que no daemon inetd.

only_from -> Define quais computadores têm permissão de utilizar o servidor xinetd.

no_access -> Define quais computadores não têm permissão de utilizar o servidor xinetd.

socket_type -> Define o tipo de soquete utilizado pelo servidor.

protocol -> Define o tipo de protocolo utilizado pelo servidor.

wait -> Define se o servidor é single thread ou multithread, geralmente no.

user -> Define o dono do servidor em questão.

server -> Define qual executável é iniciado para tratar do servidor (daemon) a ser utilizado.

server_args -> Define os parâmetros a serem passados para o servidor quando este é iniciado.

É possível migrar do superdaemon inetd para o superdaemon xinetd utilizando o comando inetdconvert, que converte os servidores configurados no arquivo /etc/inetd.conf em servidores configurados em arquivos individuais no diretório /etc/xinetd.d.

Exemplos:

#inetdconvert -d /etc/xinetd.d -inetdfile /etc/inetd.conf ftp
Converte o servidor FTP.

 # inetdconvert -d /etc/xinetd.d -inetdfile /etc/inetd.conf -convertremaining
Converte todos os servidores.

O comando inetdconvert só converte servidores cujas entradas (linhas) não estão comentadas, ou seja, não começam com o caractere "#" no arquivo /etc/inetd.conf.

/etc/hosts.allow e / etc/hosts.deny

Através destes arquivos podemos bloquear e permitir o acesso de um ip ou mais aos serviços do sistema, sendo que, o arquivo /etc/hosts.allow é usado para permitir o acesso e o arquivo / etc/hosts.deny para não permitir o acesso. O arquivo /etc/hosts.allow sempre terá preferência em relação ao /etc/hosts.deny, se um serviço estiver bloqueado para tal ip no hosts.deny e estiver libera do no /etc/hosts.allow o ip em questão conseguirá acesso ao serviço.

Vejamos um exemplo de configuração para o arquivo /etc/hosts.deny

sshd:ALL -> bloqueia o acesso via ssh para todos os ips
in.telnetd:ALL -> bloqueia o acesso via telnet para todos os ips

/etc/HOSTNAME

Armazena o nome do computador.

Exemplo do arquivo /etc/hostname

#cat /etc/hostname
firewall.raylinux.com

/etc/hosts

Contém uma lista de endereços IP e nomes de computadores da rede local.

Exemplo do arquivo /etc/hosts:

#vi /etc/hosts

127.0.0.1	**localhost.localdomain**	**localhost**
192.168.1.1	**firewall.raylinux.com**	**firewall**
192.168.1.2	**ns.raylinux.com**	**ns**
192.168.1.3	**maquina3.raylinux.com**	**maquina3**

/etc/networks

Contém uma lista dos nomes das redes de computadores e dos endereços IP.

Exemplo do arquivo /etc/networks:

#vi /etc/networks

loopnet 127.0.0.0
localnet 192.168.1.0

/etc/host.conf

Informa quais serviços utilizar para solucionar os nomes de computadores e em que ordem. A opção multi determina que um computador pode ter múltiplos endereços IP.

Exemplo do arquivo /etc/host.conf:

#vi /etc/host.conf

order hosts,bind
multi on

/etc/resolv.conf

Nesse arquivo é configurado o cliente DNS, que contém o nome do domínio do servidor de DNS e seu endereço IP.

Exemplo do arquivo /etc/resolv.conf:

#vi /etc/resolv.conf
search raylinux.com
nameserver 192.168.1.100

/etc/sysconfig/network

No Red Hat Linux e seus derivados, define o nome do computador, o nome do domínio (DNS), o nome do domínio NIS, o roteador e se a rede será ativada ou não na inicialização do Linux. Em outras distribuições, esse arquivo pode ter outro nome, formato ou localização.

Exemplo do arquivo /etc/sysconfig/network:

#vi /etc/sysconfig/network

NETWORKING=yes
HOSTNAME=workstation1.linux.org.br
DOMAINNAME=linux.org.br
GATEWAY=192.168.1.1
GATEWAYDEV=eth0

NISDOMAIN=intranet

/etc/sysconfig/network-scripts/ifcfg-lo

Define o dispositivo da interface de loopback, o seu endereço IP, a sua máscara de rede, o seu endereço de rede, o seu endereço de broadcast e se esta é ativada ou não na inicialização do Linux. Em outras distribuições, esse arquivo pode ter outro nome, formato ou localização.

Exemplo do arquivo /etc/sysconfig/network-scripts/ifcfg-lo:

#vi /etc/sysconfig/network-scripts/ifcfg-lo

DEVICE=lo
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
BROADCAST=127.255.255.255
ONBOOT=yes

/etc/sysconfig/network-scripts/ifcfg-eth0

Define o dispositivo da interface de rede, o seu endereço IP, a sua máscara de rede, o seu endereço de rede, o seu endereço de broadcast e se a mesma é ativada ou não na inicialização do Linux. Em outras distribuições, esse arquivo pode ter outro nome, formato ou localização.

Exemplo do arquivos /etc/sysconfig/network-scripts/ifcfg-eth0:

#vi /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0
IPADDR=192.168.1.2
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
ONBOOT=yes

É possível instalar várias interfaces de rede Ethernet em um computador com o Linux e para cada uma delas devemos ter um arquivo de configuração. Por exemplo:
/etc/sysconfig/network-scripts/ifcfg-eth0 é o arquivo de configuração da primeira interface de rede Ethernet,
/etc/sysconfig/network-scripts/ifcfg-eth1 é o arquivo de configuração da segunda interface de rede Ethernet, e assim por diante.

/etc/rc.d/rc.local

É o último script a ser executado, sendo possível incluir comandos ou scripts adicionais neste. Por exemplo, caso seja desejado inicializar servidores adicionais. Por padrão /etc/rc.d/rc.local simplesmente

cria uma mensagem de acesso ao sistema com a versão do kernel e o tipo da máquina. Em outras distribuições, esse arquivo pode ter outro nome ou localização.

Principais comandos de redes

Comando Descrição

ifconfig -> Exibe ou manipula a configuração das interfaces de rede.
netstat -> Exibe o status da rede.
ping -> Indica se um computador remoto pode ser alcançado.
nslookup -> Questiona o serviço de nome de domínio DNS.
traceroute -> Traça a rota percorrida por pacotes até o computador de destino.
route -> Exibe ou manipula a tabela de roteamento IP.
arp -> Manipula o cache ARP do sistema.
rarp -> Manipula a tabela RARP do sistema.

Inicializando os serviços básicos da rede
Execute os comandos:

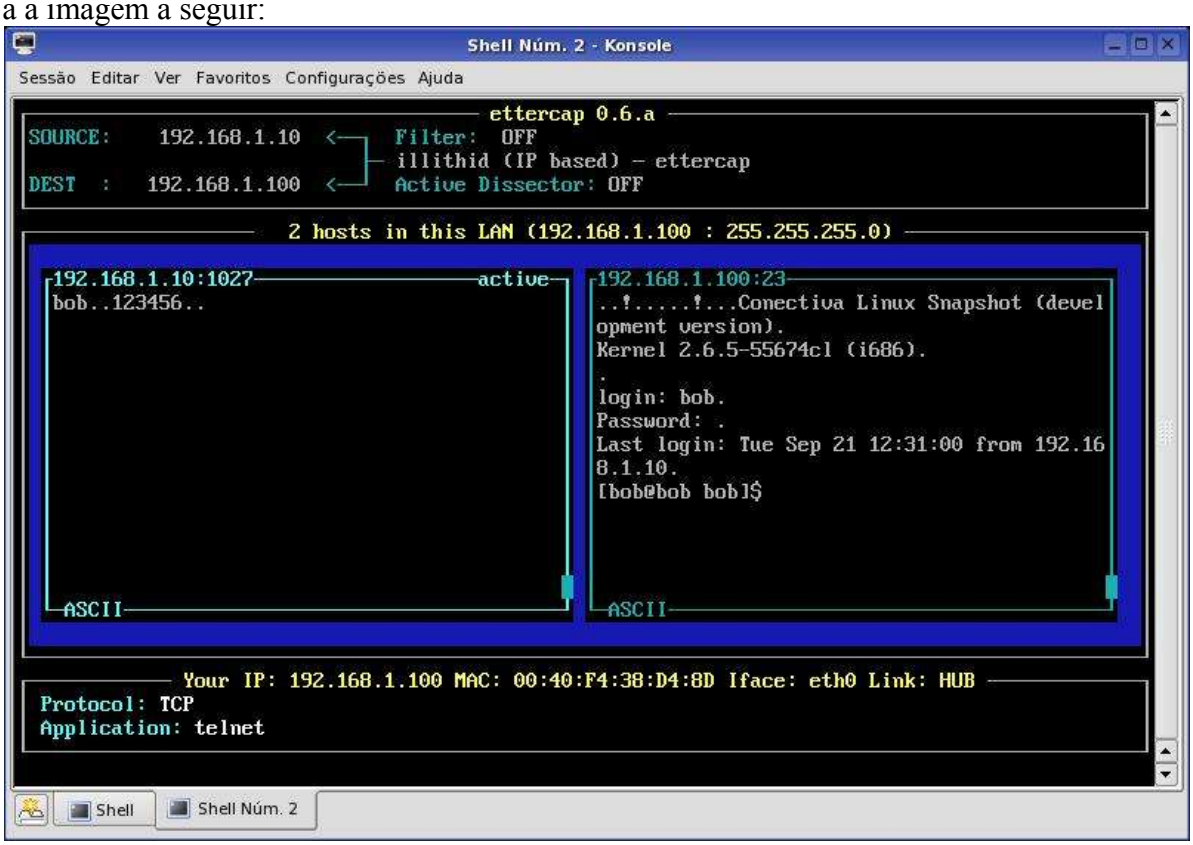
```
# service network start
# service netfs start
# service portmap start
# service xinetd start
```

Acesso Remoto

TELNET

O telnet é um programa para acesso remoto que já foi bastante utilizado. Hoje em dia dificilmente utilizamos o telnet para ascessar outra maquina com o objetivo de fazer alguma manutenção, isso por que ele não é um serviço seguro pois não trabalha com criptografia, sendo assim seus pacotes podem ser capturados e vistos facilmente por algum Sniffing.

Veja a imagem a seguir:



Pacotes necessários para instalação:

telnet-x.x -> cliente telnet para podermos logar remotamente no servidor telnet.
telnet-server-x.x -> servidor telnet que permite login remoto dentro da máquina em que ele está rodando.

Verifique se ele já está instalado.
rpm -qa | grep telnet

Caso não esteja instalado, faça a instalação da seguinte forma:

Debian – Conectiva

```
# apt-get install telnet
# apt-get install telnet-server
```

Red-Hat-Conectiva

```
# rpm -ivh telnet-x.x-xx-i386.rpm
# rpm -ivh telnet-server-x.x-xx.i386.rpm
```

Configurando o Servidor Telnet

Vamos habilitar o serviço no superdaemon xinetd.

```
# vi /etc/xinetd.d/telnetd
```

Modifique a seguinte linha:

```
disable = no
```

Agora devemos parar e subir o xinetd:

```
# service xinetd stop
# service xinetd start
```

Conectando-se no servidor telnet

Para se conectar no servidor telnet, precisamos apenas do client instalado na nossa máquina, depois e só entrar com o seguinte comando:

```
# telnet ip do servidor telnet
# telnet 192.168.1.100
```

O telnet irá pedir o usúario e a senha que estão armezanadas /etc/passwd do servidor telnet. Após feito o login entre com comandos.

Para sair basta apenas digitar **exit** .

SSH

O SSH é um programa para acesso remoto, que ao contrário do telnet tem uma conxeção segura, ou seja provem um canal de comunicação seguro entre duas maquinas ou hosts, difcultandoo o compreendimento de programas de sniffing. Ele substitui o rlogin e rsh e implementa suporte ao secure shell protocol.

Pacotes necessários para a instalação do SSH:

openssh-x.x -> contém ssh-keygen, scp.
openssh-clients-x.x -> contém os clientes necessários para fazer conexões com servidores ssh.
openssh-server-x.x -> contém o servidor ssh.

Debian – Conectiva

 # apt-get install openssh
 # apt-get install openssh-clients
 # apt-get install openssh-server

ou

 # rpm -ivh openssh-x.x -x.x-xx-i386.rpm
 # rpm -ivh openssh-clients-x.x-x.x-xx.i386.rpm
 # rpm -ivh openssh-server-x.x-x.x-xx.i386.rpm

Configurando o Servidor SSH

Primeiro vamos editar o arquivo de configuração do servidor que está **/etc/ssh/ssh_config** .

 # vi /etc/ssh/ssh_config

PermitRootLogin no -> para que ninguém se log como root no meu servidor SSH
PermitEmptyPasswords no -> para não permitir criação de senhas vazias
PasswordAuthentication yes -> para habilitar autenticação de senhas

Obs: Para versões do openssh-server-3.8 editamos apenas a linha PasswordAuthentication.

Depois de editar o arquivo, vamos subir o daemon do servidor ssh.

 # service sshd stop -> se caso ele esteje rodando, se não pule para o próximo passo.
 # service sshd start

Configuração do Cliente SSH

Para ser cliente de um servidor ssh, você deve ter o pacote openssh-clients instalado, sem a necessidade de se ter o daemon do sshd rodando ou mesmo o pacote do openssh-server. Veja os comandos para se conectar num servidor ssh.

#ssh hostname -> conecta-se ao servidor ssh pelo nome de hostname, apenas como root.
#ssh raylinux.com

#ssh usuario@hostname -> conecta-se ao hostname do servidor ssh com um determinado usuário.
#ssh bob@raylinux.com

#ssh usuario@ip -> conecta-se ao IP do servidor ssh com um determinado usuario.
#ssh bob@192.168.1.100

A primeira vez que se conectar a um servidor ssh ele irá perguntar se você deseja continuar, está pergunta é feito apenas uma vez, pois ele irá fazer o fingerprint, ou seja a troca de chaves para reconhecimento. Da próxima veze que tentar se conectar novamente ele não irá mais fazer está pergunta.

SCP

O SCP (Segure Copy), é uma outra funcionalidade boa do openssh-clients, com ele você pode estar fazendo cópias seguras de arquivos e diretórios remotamente, veja a seguir os comandos usados:

Cópia de Arquivos

#scp usuario@IP:/diretorioremoto/arquivo diretoriolocal
#scp bob@192.168.1.100:/tmp/aula.doc /home/bob

Conecta-se como usuario bob ao servidor ssh IP 192.168.1.100 no diretorio /tmp/ e faz o download do arquivo aula.doc para o seu diretório /home/bob.

#scp arquivo usuario@IP: diretorioremoto
#scp rwindows.doc bob@192.168.1.100:/tmp

Conecta-se como usuario bob ao servidor ssh 192.168.1.100 enviando ou fazendo o upload do arquivo rwindows.doc para o diretorio /tmp

Cópia de Diretório

#scp –r usuário@192.168.1.100:/diretorioremoto diretóriolocal
#scp -r bob@192.168.1.100:/tmp/ /tmp

Conecta-se como usuario bob ao servidor ssh IP 192.168.1.100 e cópia o diretorio /tmp e toda sua

estrtura para o seu diretório /tmp.

#scp –r diretoriolocal/ ip:/diretórioremoto
#scp -r /tmp/ bob@192.168.1.100:/tmp

Conecta-se como usuario bob ao servidor ssh IP 192.168.1.100 e faz um upload do diretório /tmp/ e toda sua estrura de diretorios para o diretório /tmp .

Servidor DNS

O DNS (Domain Name Service), tem como sua principal função resolver nomes, ou seja converter IP para hostname e vice versa.

Sendo mais prático o que o servidor DNS faz?

Exemplo:

Quando utilizamos um browser (Internet Explorer, Mozilla, Konqueror), para ascessar um determinado site na internet por exemplo www.raylinux.com, estamos acescessando na verdade o servidor DNS desta empresa, onde o nome e o DNS está registrado na InterNIC (Orgão detentor dos dominios .com), este servidor tanto pode estar hospedado na propria empresa como em algum um outro provedor de hospedagem. A função do servidor DNS é resolver o endereço www.raylinux.com para um numero IP, que no nosso caso seria 216.52.184.240 que é um endereço válido na Internet.

Imagine você o que seria a Internet sem o DNS, ao invés de decoramos www.raylinux.com teriamos que que decorar http://216.52.184.240.

Pode-se também utilizar o Servidor DNS para uso interno, ou seja sem necessitar de um registro e um IP válido na Internet.

InterNIC: é um serviço do departamento de comércio do governo dos Estados Unidos que detinha o monopolio dos dominios .com, .net e .org, por um preço de US$70,00.

Com a quebra do monopolio da InterNIC estes dominios já estão disponiveis por outras empresas registradoras, inclusive nacionais com um preço bem mais baixo chegando a variar entre R$40,00 a R$45,00 anuais.

FAPESP: Fundação de Amparo à Pesquisa do Estado de São Paulo, é um órgão governamental responsável pelo registro e manutenção dos domínios .br .

Pode-se também utilizar o Servidor DNS para uso interno, ou seja sem necessitar de um registro e um IP endereço válido na Internet.

Obs: O exemplo dado seria no caso da empresa ter o registro em algum orgão detentor de dominios, haveria a necessidade de informamos dois DNS com IP válidos na Internet no caso o DNS primário e o DNS Secundário comos veremos a seguir.

Servidor Primário: É a fonte das informações consultadas, tem total autoridade de responder por uma informação questionada.

Servidor Secundário: Transfere informações a partir do Servidor Primário, ou seja mantém uma cópia do mesmo.

BIND

Bind (Berkeley Internet Name Domain), o bind inclui o servidor DNS que é conhecido como named. O cliente DNS é conhecido como solucionador ou resolvedor, quando um computador precisa saber a qual endereço ip um nome se refere, solicita a resolução de nomes ao servidor DNS.

Configurando o servidor DNS.

Pacotes necessários:

bind-x.x-xx -> servidor DNS e arquivos de configurações.
Bind-utils-x.x-xx -> comandos do DNS.

 # apt-get install bind
 # apt-get install bind-utils

ou

 # rpm -ivh bind-x.x -x.x-xx-i386.rpm
 # rpm -ivh bind-utils-x.x-x.x-xx.i386.rpm

Arquivos de Configurações

/etc/named.conf	-> apontamento das zonas de DNS
/var/named/primario/dominio.conf	-> extensão do apontamento das zonas do DNS
/var/named/zona/db.dominio	-> base de dados do DNS
/var/named/zona/rev.dominio	-> arquivo de reverso do DNS
/var/named/named.ca	-> apontamento para o rootserver
/etc/resolv.conf	-> como já visto e o apontamento para os servidores que resolverão os nomes
/etc/host.conf	-> responsável pela ordem de busca em uma estação.
/etc/nsswitch.conf	-> Arquivo responsável por agilizar o acesso as informações do servidor DNS

Antes de começar criaremos os diretórios do nosso dominio primário e a zona ond estarão os arquivos reverso e base de dados.

#mkdir /var/named/primario
#mkdir /var/named/zona

named.conf

Vamos começar a entender e editar o arquivo /etc/named.conf .

#vi /etc/named.conf

Segue abaixo um exemplo do arquivo named.conf

```
// named.conf
// generated by named-bootconf.pl

options {
        directory "/var/named";
};

//
// a caching only nameserver config
//
controls {
        inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};

zone "." {
        type hint;
        file "named.ca";
};

zone "localhost" IN {
        type master;
        file "localhost.zone";
        allow-update { none; };
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "named.local";
        allow-update { none; };
};

//include "/etc/rndc.key";
include "primario/raylinux.conf";
```

directory -> especifica o diretório onde será feita toda estrura das zonas e base dados.
hint -> especifica a zona de cache no caso o named.ca.
master -> especifica a zona do dominio primário.
slave -> específica a zona do domonío secundário.
include -> faz o apontamento de uma inclusão de outra zona, que está configurando em outro dominío no nosso caso /var/named/primario/dominio.conf

Aqui apenas faremos a inclusão que aponta para outro arquivo de configuração dominio.conf, neste caso raylinux.conf.

dominio.conf

Vamos criar e configurar as informações do nosso domínio, nele estará também o apontamento para a nossa base de dados e o reverso.

#vi /etc/named/primario/dominio.conf

Exemplo de um dominio.conf, neste caso raylinux.conf :

#vi /var/named/primario/raylinux.conf

```
// raylinux.conf

// dominio e apontamento para o base de dados

zone "raylinux.com" {
        type master;
        file "zona/db.raylinux";
};

// apontamento do reverso e ip do servidor master

zone "100.1.168.192.in-addr.arpa" {
        type master;
        file "zona/rev.raylinux";
};
```

db.dominio e rev.dominio

No diretório /var/named deverá conter os seguintes arquivos e diretórios:

#ls /var/named

dump localhost.zone named.ca named.local primario zona

localhost.zone -> base de dados localhost
named.local -> é utilizado como base para a construção da basa de dados e o reverso.Vamos configurar a base de dados e o reverso.

Copie o arquivo localhost.zone e named.local para o diretório /var/named/zona/ para podermos aproveitar suas estruturas, o que facilita em muito e depois vamos renomear para db.dominio e rev.dominio.

#cp -r /var/named/localhost.zone /var/named/zona/
#mv /var/named/primario/localhost.zone /var/named/zona/db.dominio.conf

#cp -r /var/named/named.local /var/named/zona/
#mv /var/named/primario/named.local /var/named/zona/rev.dominio.conf

Agora vamos editar o db.dominio e o rev.dominio

#vi /etc/named/zona/db.dominio -> troque db.dominio pelo seu dominio

Exemplo de um db.dominio, neste caso db.raylinux:

#vi /var/named/zona/db.raylinux

```
@       IN    SOA     ns.raylinux.com.  root.raylinux.com.  (
                              1     ; Serial
                              8H    ; Refresh
                              2H    ; Retry
                              1W    ; Expire
                              1D)   ; Minimum

@                     IN      NS      ns.raylinux.com.
@                     IN      MX 10   mail.raylinux.com.
raylinux.com.         IN      A       192.168.1.100
ns                    IN      A       192.168.1.100
www                   IN      CNAME   raylinux.com.
ftp                   IN      CNAME   raylinux.com.
mail                  IN      CNAME   raylinux.com.
```

#vi /etc/named/zona/rev.raylinux -> troque rev.dominio pelo seu dominio

Exemplo de um rev.dominio, neste caso rev.raylinux:

```
@       IN      SOA     ns.raylinux.com. root.raylinux.com.  (
                          1         ; Serial
                          8H        ; Refresh
                          2H        ; Retry
                          1W        ; Expire
                          1D)       ; Minimum

100.1.168.192.in-addr.arpa.      IN      NS      ns.raylinux.com.
100.1.168.192.in-addr.arpa.      IN      MX      mail.raylinux.com.
100.1.168.192.in-addr.arpa.      IN      PTR     www.raylinux.com.
```

Registros DNS

@	-> define o nome da zona.
SOA	-> define a zona que será autoridade.
IN	-> classe=Internet.
NS	-> lista um servidor DNS para este dominio.
A	-> mapeamento de nomes para endereços.
MX	-> servidor de e-mail deste dominio.
PTR	-> mapeamento reverso, ou seja IP para nome.
CNAME	-> nomes canônicos.
ns.raylinux.com	-> hostname do servidor.
root.raylinux.com	-> e-mail do administrador.
Serial	-> define as alterações do banco de dados, importante para a atualização do servidor secundário.
Refresh	-> especifica o tempo em que o servidor secundário irá consultar o servidor primário para rever as alterações e atualiza-las.
Retry	-> especifica o tempo em que o servidor secundário irá tentar atualizar o banco de dados no caso de falha.
Expire	-> especifica o tempo em segundos que pode decorrer até que o servidor secundário considere seus dados desatualizados, sem fazer uma atualização.
Minimum	-> especifica o tempo padrao para o servido exportar os registros de dominio.

named.ca

Aqui conterão os hosts e IPs dos rootservers da InterNIC, se caso você não utilize o servidor DNS não há necessida de se ter este arquivo então apague toda as linhas, mas antes faça um backup dele, caso contrário deixe como está, não o modifique.

resolv.conf

Enfim estamos quase lá.
Configure o arquivo /etc/resolv.conf

#vi /etc/resolv.conf

search dominio.com.br	-> dominio no qual pertence o host.
nameserver IP	-> IP do servidor DNS.

Exemplo:

#vi/etc/resolv.conf
search raylinux.com

nameserver 192.168.1.100

Terminado todas as configurações, vamos subir o daemon do servidor DNS

#service named stop
#service named start

Testando o Servidor DNS

Para testarmos o Servidor DNS utilizaremo o comando dig:

#dig dominio.com.br ANY -> mostra as informações sobre os servidores dns

#dig dominio.com.br MX -> mostra informações sobre os servidores de e-mail

SERVIDOR FTP

O FTP é utilizado de forma personalizada e automática em soluções que trabalham como o EDI (Eletronic Data Interchange), onde Matrizes e Filiais trocam arquivos de dados com a finalidade de sincronizar seus bancos de dados. Outro uso seria os LiveUpdates, como o usado nas atualizações dos produtos da Symantec (Norton Antivírus, Personal Firewall e etc.).

Existem também os programas que aceleram download e que utilizam o protocolo FTP. Esses programas usam tecnologia de múltiplas sessões e empacotamento com a quebra dos arquivos, conseguindo dessa forma, uma melhora significativa na velocidade dos downloads.

Os modos de transferência em detalhes

Padrão

No modo padrão a primeira conexão que é estabelecida pelo cliente é em uma porta TCP de número alto (varia entre 1024 a 65535, pois é dinâmica) contra o servidor na porta TCP número 21. Essa conexão é quem autentica e diz ao servidor qual(is) arquivo(s) o cliente deseja. Esta conexão permite também, a passagem de outras informações de controle (comandos por exemplo). Contudo, quando chega à hora de transferir os dados reais uma segunda conexão será aberta. Diferente da conexão de controle, esta que é de dados, é aberta pelo servidor em sua porta TCP de número 20 contra o cliente em uma porta TCP de número alto e que é atribuída também dinamicamente (cliente e servidor negociam a porta em questão como parte da troca da conexão de controle).

Passivo

No modo passivo a primeira conexão é idêntica ao modo padrão. Contudo, quando chega à hora de transferir os dados reais, a segunda conexão não opera da mesma forma que no modo padrão. Ela opera da seguinte forma: o servidor fica esperando que o cliente abra a conexão de dados. Essa conexão e aberta pelo cliente em uma porta TCP de número alto (varia entre 1024 a 65535, pois é dinâmica) contra o servidor em uma porta TCP de número alto também. Tudo fica estabelecido na conexão de controle inclusive a porta TCP que o cliente vai usar contra o servidor. Além de modificar o sentido da conexão de dados, as portas são altas em ambos os lados.

PROFTP

O PROFTP, é um aplicativo utilizado para nos conectarmos a um servidor ftp e executarmos download e upload, da mesma maneira que o telnet, a comunicação entre clientes e servidor é feita em clear text, sem qualquer tipo de criptografia.

Pacotes necessários para instalação:

ftp-x.x -> cliente ftp
proftpd-x.x -> servidor ftp

Verifique se ele já está instalado.

 # rpm -qa | grep telnet

Caso não esteje, instale-o da seguinte forma:

 # apt-get install ftp
 # apt-get install proftpd

ou

#cd /mnt/cdrom/conectiva/RPMS
#rpm -ivh ftp-x.x-xx-i386.rpm
#rpm -ivh proftpd-x.x-xx.i386.rpm

Principais arquivos e diretórios.

/etc/proftpd.conf -> arquivo de configuração do proftpd.
/etc/ftpusers -> arquivo de usuários que não podem acessar o ftp.
/srv/ftp -> diretório de arquivos dos usúarios anonymous.
/var/ftp/pub -> diretório de arquivos dos usúarios anonymous nas versões antigas do proftpd

Configurando o Sevidor FTP

Vamos então configurar o proftpd, acesse o arquivo de configuração /etc/proftpd.conf .

#vi /etc/proftpd.conf

ServerName **"ProFTPD - Default Instalation"** -> nome do servidor
ServerType **standalone** -> utilizaçao do daemon proftpd
ScoreboardFile **/var/run/proftpd/scoreboard** -> arquivos de armazenamentos dos pid
ServerAdmin **root@localhost** -> e-mail do administrador

SyslogFacility	**AUTH**	-> prioridade do syslog
Port	**21**	-> porta padrão
Umask	**022**	-> permissão padrão
MaxInstances	**30**	-> maximo de usuarios conectados
User	**proftpd**	-> usuario que irá executar processos
Group	**proftpd**	-> grupo que irá executar processos

Salve o arquivo e suba o daemon do servidor.

Verifique a lista dos usuários que por padrão não poderão se conectar pelo ftp.

#vi /etc/ftpusers

Agora é só subir o daemon.

#service proftpd stop
#service proftpd start

Usuários Anonymous

Por padrão o proftpd comenta as linhas que permitem usuarios anonymous a se conectar no servidor ftp veja a seguir um exemplo de suas configurações básicas:

```
<Anonymous /srv/ftp>
  User                    ftp
  Group                    ftp
  UserAlias               anonymous ftp
  MaxClients              10 "Sorry, max %m users -- try again later"
  DisplayLogin            welcome.msg
  DisplayFirstChdir        .message
  AccessGrantMsg          "Anonymous access granted for %u."
<Limit WRITE>
  DenyAll
 </Limit>
</Anonymous>
```

Cliente FTP

Se conecte no servidor ftp do colega, utilizando a conta de um usuário qualquer que tenha no servidor.

#ftp IP
#ftp 192.168.1.100

Verificando se a conexão do servidor ftp foi bem sucedida

ftp> status

Fazendo o download de um arquivo do servidor ftp, para o diretório corrente na máquina cliente

ftp> mget arquivo

Faça o upload de um arquivo do diretório local da máquina cliente e mande para o servidor

ftp>put arquivo

Servidor DHCP

O objetivo de um servidor dhcp é prover para as estações da rede as configurações TCP-IP (endereço ip, endereço do servidor DNS, endereço de rede, etc) de forma automática. Normalmente, iremos utilizar este serviço em redes com um número muito grande de máquinas.

Os pacotes necessários para trabalhar com o dhcp são:

dhcpcd -> daemon do servidor dhcp
dhcp -> servidor e cliente dhcp

Configuração do servidor DHCP:

Edite o arquivo de configuração /etc/dhcp.conf -> este arquivo define como o servidor dhcp, irá atribuir a configuração TCP-IP para as estações. Exemplo de configuração do arquivo:

```
#configuração do servidor dhcp
        subnet 192.168.1.100 netmask 255.255.255.0 {
          range 192.168.1.2 192.168.1.30;
          default-lease-time 600;
          max-lease-time 7200;
          option subnet-mask 255.255.255.0;
          option broadcast-address 192.168.1.255;
          option routers 192.168.1.100;
          option domain-name-servers 192.168.1.100;
          option domain-name "raylinux.com";
}
```

APACHE

O Apache é o servidor Web mais utilizado na Internet, desde 1996, pois está presente em 60% dos computadores.

Pacotes necessários:

apache-x.x-xx -> servidor web, contendo o daemon e os arquivos de configuração.
openssl-x.x-xx -> suporte a ssl, ou seja paginas seguras.
php-x.x-xx -> software e modulo para programação em php.

Principais Diretórios e Arquivos

/etc/apache/conf/http.conf -> arquivo de configuração do apache.
/srv/www/default/html -> diretório onde estarão armazenados as paginas.
/usr/sbin/httpd -> daemon httpd.
/var/run/httpd.pid -> arquivo de armazenamento do pid do apache.
/etc/httpd/conf/httpd.conf -> arquivo de configuração em outras distribuições.
/var/www/html -> diretório onde são armazenadas as pagínas web em outras distribuições.

Configurando o Apache

Vamos entender e editar o arquivo de configuração /etc/apache/conf/http.conf .
Para uma configuração básica só pricisamos editar as seguintes linhas.

Listen 80 -> porta padrão onde o apache irá rodar
User www -> usuario pagrão que irá executar os processos
Group www -> grupo padrão que irá executar os processos
ServerAdmin root@raylinux.com -> email do admnistrador
ServerName raylinux.com:80 -> dominio ou ip do servidor web, seguido da porta
DocumentRoot "/srv/www/default/html" -> diretório da onde estarão as páginas
DirectoryIndex index.htm index.html.var index.html -> tipo de pagina que ele irá bsucar primeiro para abrir.

Agora vamos habilitar também suporte a **SSL.**

Segue abaixo um exemplo

<VirtualHost www.raylinux.com:443>
ServerAdmin root@raylinux.com -> email do administrador.
DocumentRoot /srv/www/default/html -> qual diretório estará as paginas com ssl.
ServerName raylinux.com -> nome do servidor.
ErrorLog /var/log/apache/raylinux.com-error_log -> diretórios de logs de erros

CustomLog /var/log/apache/raylinux.com-access_log common -> diretórios de acessos.
SSLcertificateFile /etc/apache/conf/ssl.crt/server.crt -> diretório de armazenamento do certificado digital.
SSLCertificateKeyFile /etc/apache/conf/ssl.key/server.key -> diretório de armazenamento do da chave de criptografada.
</VirtualHost>

Antes de subirmos o daemon usaremos o comando a seguir para checar algum erro na configuração.

#apachectl -t

Se não retorna nenhum erro passe adiante e suba o daemon, caso contrário verifique a linha no qual houve erro e concerte.

Agora só pricisamos o daemon do apache.

#service httpd stop
#service httpd start

Testando a configuração

Para testarmos a configuração vamos abrir o browser (Mozilla ou Konqueror), e digitarmos na barra de endereço o IP de nossa maquina, ou se caso o servidor DNS esteje rodando podemos utilizar o dominio.

Exemplos: **http://192.168.1.100**
http://www.raylinux.com

Veja se abriu a pagina do apache, caso tenha aberto tudo correu bem, se não veja se houve algum erro na configuração, ou veja se o diretorio dos arquivos das paginas estão corretos / srv/www/default/html.

Agora testaremos o SSL, digite novamente no seu browser o IP ou seu dominio caso ele esteje rodando, mas ao invés do http:// digite https:// .

Exemplo: **https://192.168.1.100**
https://www.raylinux.com

Se tudo correu bem a mesma pagina será aberta mas com uma diferença, a pagina estará utilizando um certificado e um chave criptografada, note na barra do bownser no canto inferior direito um cadeado.

Suporte para PHP4

Aos futuros administradores web e programadores php, aqui vamos configurar agora o apache com suporte a php, para executarmos paginas dinâmicas.

É necessário o pacote do php instalado como já foi feito antes a instalaçao não pricisaremos nos preocupar com isto.

Configurando o apache para suporte ao PHP4

Vamos retornar ao arquivo de configuração do apache.

 # vi /etc/apache/conf/httpd.conf

Inclua as seguintes linhas:

LoadModule php4_module /usr/lib/apache/modules/libphp4.so
AddType application/x-htpd-php4 .php4

Agora procure pela linha onde está o **DirectoryIndex** e inclua o index.php

DirectoryIndex index.php index.html index.htm

Salve o arquivo e saia.
Pare o daemon e cheque novamente as configurações dop apache.

#service httpd stop
#apachectl -t

Se tudo correu bem, suba o daemon.
#service httpd start

Testando a configuração

Vamos criar um arquivo index.php simples no diretório /srv/www/default/html .

#vi /srv/www/default/html/index.php

```
<?
    phpinfo()
?>
```

Salve o aquivo e saia.

Novamente abriremos o nosso browser e digitaremos o nosso IP ou dominio, caso ele esteje rodando.

Exemplo:
http://192.168.1.100
www.raylinux.com

Caso abra a página do php ocorreu tudo bem, executamos uma pagina php, caso contrário reveja suas configurações.

POSTFIX

Postfix é um agente de transporte de mensagens (MTA) escrito e mantido por Wietse Venema, conhecido autor dos softwares tcp-wrappers (distribuídos atualmente na grande maioria das distribuições GNU/Linux), da ferramenta para auxílio na análise de intrusões SATAN e do software de auxílio a computação forense The Coroner's Toolkit (TCT), dentre diversos outros conhecidos softwares livres voltados a área de segurança. O Postfix foi criado inicialmente para oferecer uma alternativa ao então (e ainda hoje, apesar de em uma proporção menor) mundialmente usado MTA Sendmail (http://www.sendmail.org). Ele foi pensado de forma a ser um MTA rápido, fácil de administrar e seguro, ao mesmo tempo sendo compatível o bastante com o sendmail para não assustar os usuários existentes. Foi intencionalmente desenvolvido para se parecer, na medida do possível, com o sendmail, mas internamente é completamente diferente.

Inicialmente, o Postfix era conhecido como VMailer e foi lançado no final de 1998 pela IBM com o nome de IBM Secure Mailer. A partir de então, assumiu o nome de Postfix e é assim que é conhecido até hoje.

Em primeiro lugar, baixe os pacotes do postfix de acordo com sua distribuição. Recomendo os sites:

Pacotes necessários para instalação:

postfix-x.x -> MTA servidor de e-mail
pop3-x.x ->

Configurando o Postfix

Após a instalação dos pacotes necessários, entre no diretório /etc/postfix (cd /etc/postfix):

#vi /etc/postfix/main.cf

Faça as seguinte alterações:

myhostname = mail.domain.com -> hostname
mydomain = domain.com -> dominio
myorigin = $mydomain -> dominio o qual aparecerá após o @ no e-mail.
inet_interfaces = all -> hosts que terão acesso a internet, pode ser definido também pelo seu IP
#inet_interfaces = 192.168.1.10, 192.168.1.11, 192.168.1.12
mydestination = $myhostname, localhost.$mydomain, $mydomain, mail.$mydomail,
www.$mydomain, ftp.$mydomainmydestination

Dominio o qual o servidor de e-mail será responsável. Podemos definir aqui os dominios virtuais também, apontando para algum arquivo como por exemplo:

mynetworks = 192.168.1.0/24, 127.0.0.0/8 -> Aqui defineremos qual rede nosso servidor de e-mail irá atender

Cofigurando relay no Postfix

Por padrão o Postfix permite relay apenas para rede definida no aquivo **main.cf,** vamos configurar-lo no servidor.

smtpd_recipient_restrictions=permit_mynetworks
check_client_acces hash>/etc/postfix/client_acces
check_relay-domains

ou

relay_domains = $mydomain, /etc/postfix/client_acces -> dominio que poderao fazer relay no servidor.

Vamos criar o arquivo /etc/postfix/client_acces

#vi /etc/postfix/client_acces

Agora dentro do aquivo vamos permitir ou negar relay no nosso servidor.

autorizado.com.br OK
autorizado2.com.br OK
rejeitado.com.br REJECT
microsoft.com REJECT

Perceba que depois do dominio sempre irá ter OK para validar o relay ou REJECT para rejeitar.
Agora vamos criar o mapa para os clientes acessarem o servidor de e-mail.

#postmap /etc/postifix/client_access

Aliases

Os alias nada mais são que apelidos que podemos usar para os nossos e-mails.
Exemplo de um alias:

postmaster: root

O e-mail root@dominio.com também receberá e-mails como postmaster@dominio.com, isto é muito útil para camuflarmos nosso e-mail original.

Para habilitarmos os aliases, abra o o arquivo de configuração do postfix /etc/postfix/main.cf.

#vi /etc/postfix/main.cf

descomente a linha:

alias_database = hash:/etc/postfix/aliases

Salve e saia do arquivo.

Agora acesse o arquivo de aliases /etc/postfix/aliases.

#vi /etc/postfix/aliases

Crie um alias para o root como nome admin.

admin: root

Saia e salve o arquivo, depois vamos atualizar o aliases com o seguinte comando:

#postalias /etc/postfix/aliases

Aqui termina a configuração do arquivo main.cf . Feita estas alterações, deve-se reinicializar o Postfix. Para efetuar esta ação utilizamos os seguintes comandos e parâmetros:

 # service postfix stop
 # service postfix start

POP3

Para podermos receber mensagens devemos configurar e carregar no superdaemon xinetd o pop3. Edite o arquivo /etc/xinetd.d/pop3

#vi /etc/xinetd.d/pop3

```
service pop-3
{
        disable     = no
        flags       = REUSE
        socket_type = stream
        protocol    = tcp
        wait        = no
        user        = root
        server      = /usr/sbin/ipop3d
}
```

Agora vamos subir o superdaemon xinetd.

#service xinetd stop

#service xinet start

Testando o Postfix

Para testarmos o Postfix utilizaremos o telnet como já foi visto antes o telnet e pouco utilizado nos dias atuais para manutenções remotas, porém muito útil para testarmos configurações.

Digite os seguinte comando:

#telnet IP 25
#telnet 192.168.1.100 25

#telnet IP 110
#telnet 192.168.1.100

Caso retorne o banner do Postfix, tudo ocorreu bem. No segundo caso, se retornar OK, está funcionando perfeitamente.

SAMBA

O samba é o programa que permite a integração entre máquinas Windows e Linux, podendo ser usado tanto como cliente e servidor. Hoje por ter um desempenho e níveis de segurança maiores vem sendo usado em muitos servidores de arquivos.

Pacotes necessários para instalação:

samba-clients samba-x.x -> clientes samba
samba-server-x.x -> servidor samba
samba-common -> ferramentas do samba

Principais Diretórios e Comandos

/etc/smb.conf -> arquivo de configuração do samba.
smbpasswd -> comando que irá gerar a senha para úsuarios smb.
smbclient -> cliente SMB, comando que verificará os compartilhamentos ativos.
smbprint -> comando para envio de impressão no smb.
smbstatus -> comando que apresenta a situação atual das conexões SMB.
testparm -> comando que verifica o arquivo smb.conf.
mount -> comando utilizado para montar compatilhamento smb
umount -> comando utilizado para desmontar compartilhamento smb.

Antes de começarmos crie dois diretórios no /tmp chamado samba1 e samba2 e de permissão para eles de escrita e leitura.

#mkdir /tmp/samba1 /samb2
#chmod 755 /tmp/samba1 samba2

As configurações do samba ficam em /etc/samba/smb.conf, abra este arquivo.

#vi /etc/samba/smb.conf

[global] -> Define as configurações globais do SAMBA

workgroup = MYGROUP -> Especifica o Domínio ou Workgroup a que o Host pertence na Rede.
server string = Conectiva Linux SMB Server -> Comentário para este Host na Rede.
printcap name = cups -> Indica o arquivo para busca das definições das impressoras.
load printers = yes -> Disponibiliza as impressoras para a rede.
printing = cups ->Indica qual o sistema de impressão padrão utilizado
log file = /var/log/samba/log.%m -> arquivo onde armazenará os logs do smb.
debug level = 1 -> Permite ao SAMBA trabalhar corretamente com algumas situações de erro.
security = user -> As permissões são dadas de acordo com o login do usuário, ou através dos grupo.
encrypt passwords = yes -> Permite senha criptografada

smb passwd file = /etc/samba/smbpasswd -> arquivo que contém as senhas dos uúarios

[homes] -> Define as configurações
comment = Home Directories -> Comentário para este compartilhamento
browseable = no -> Define se o compartilhamento será ou não visível para o Ambiente de Rede.
writable = yes -> Indica se o usuário poderá ou não escrever em sua pasta pessoal

[printers] ->Define as configurações da impressora
comment = All Printers -> Comentário para a impressora
path = /var/spool/samba -> defina o diretório daFila de impressão no smb.
browseable = no
guest ok = no
writable = no
printable = yes
printer admin = root -> admnistrador da impressora

Usando o nível de segurança como **SHARE**, compartilhe o seu /tmp/samba1 .

Security = SHARE
comment = Past do Samba1
path = /tmp/samba1
public = yes
browseable = yes
writeable = yes
read only =no

Usando o comando **testparm** verifique se o arquivo de configuração contem algum erro.

#testparm /etc/samba/smb.conf

Levante os daemons smbd.

#service smb start

Verifique se os seus compartilhamentos estão ativos.

#smbcliente –L meu_IP –N

Verifique os compartilhamentos das outras máquinas.

#smbcliente –L IP_remoto –N

Monte um compartilhamento do diretório samba1 em /tmpb/samba2, e ignore o pedido de senha.

#**mount –t smbfs //IP_remoto/samba1 /samba**

Edite novamente o /etc/samba/smb.conf mudando o nível de segurança para **user** e colocando restrição de usuário em um dos seus compartilhamentos.

Security = USER
comment = Pasta de troca
path = /tmp/samba1
public = yes
browseable = yes
writeable = yes
read only = no
valid users = usuario

Adicione um usuario existentes no sistema, no Samba e reinicialize o samba.

#**smbpasswd –a usuario**

#service smb restart

Agora precisamos definir que usuario vai montar o compartilhamento do Samba durante a montagem, monte um compartilhamento e no momento da senha coloque a senha definida no samba.

#mount –t smbfs //IP_remoto/tmp/samba1 /tmp/samba2 –o username=usuario

NFS

O NFS (Network File System)Permite que você compartilhe sistemas de arquivos entre computadores. O acesso aos arquivos montados pela rede é completamente transparente para o usuário. Os principais usos do NFS são para centralizar informações acessados por muitas pessoas, armazenar arquivos muito grandes facilitando o backup da rede.

Verifique se você têm os pacotes para NFS instalados.

#rpm –qa | grep nfs

Principais Diretórios e comandos

/etc/exports -> arquivo de configuração do nfs responsável por controlar os diretórios e os tipo de acesso permitido.

mount -> comando utilizado para montar os sistemas de arquivos.
umount -> comando utilizado para desmontar os sistemas de arquivos.
exportfs -> comando utilizado para processar e atualizar o arquivo /etc/exports.

Configurando um Servidor NFS

Crie dois diretórios e dê a permissão a 775 para eles.

#mkdir /share1 /share2
#chmod 755 /share1 /share2

Especificamos os diretórios que queremos compartilhar no arquivo /etc/exports. Vamos compartilhar o /pasta1 para todos com permissão de leitura e escrita e a /pasta2 também com as mesmas permissões só que apenas para um usúario

#vi /etc/exports

/pasta1 * (rw)
/pasta2 192.168.1.x/24 (rw)

Compartilhe o home de um usuário com permissão de escrita para apenas uma maquina e leitura para o restante da rede.

/home/usuario 192.168.1.x (rw,insecure,no_root_squash)
192.168.1.0/24 (ro,secure,root_squash)

A opção **secure** só aceita pedidos de conexão feitas por portas abaixo da 1024, e a opção **insecure** faz o inverso.
A opção **root_squash** faz com que o superusuário seja considerado com usuário comum,

normalmente **nobody**, e a opção **no_root_squash** permite uid=0 nas conexões.

Os daemons usados pelo NFS são rpc.mountd e rpc.nfsd.

Inicialize os daemons do servidor NFS.

#service nfsd start
#service nfsd stop

Agora verifique com o comando **showmount** os compartilhamentos que estão ativos na sua máquina e nas máquinas remotas.

#showmount –e meu_ip
#showmount –e ip_remoto

#showmount –e 192.168.1.10
#showmount –e 192.168.0.20

Configurando o Cliente NFS

Visto os compartilhamentos que estão ativos com o comando showmount monte um compartilhamento remoto, ou seja de um colega ao lado.

mount –t nfs 192.168.0.20:/pasta1 /pasta2

Veja que o acesso aos arquivos de host remoto é transparente, e para desmontarmos usamos o comando **umount** .

Com o comando exportfs você pode adicionar ou remover compartilhamentos sem precisar reinicializar o serviço. Remova o compartilhamento do /pasta1

#exportfs –u *:/pasta1

Visualiza novamente os seus compartilhamentos.

#showmount –e 192.168.0.10

Reative o compartilhamento do /tmp sem reinicializar o serviço.

#exportfs –a –r

NIS

O NIS (Network Information Service) é um serviço desenvolvido pela SUN com a finalidade de disseminar informações de uma rede. Informações estas como os grupos de usuários, usuários, hosts e etc.

No inicio o NIS era conhecido com YP (yellow page), daí sempre as inicias YP antes dos principais serviços do NIS (Ex.: ypbird, ypcap), porém por um problema de patente e marca com a British Telecom, o nome acabou mudando.

O uso do NIS é muito importante em redes com a qual tem se um ambiente com várias máquinas. Isso é importante para assegurar aos usuários acessos a estas maquinas de uma forma uniforme, ou seja, garantir os mesmos níveis de acesso em todas as máquinas.

Existem três principais tipos de NIS:
NIS2 - É a versão original, ainda conhecida como Yellow Page.
NISY - É uma versão com vários aplicativos de configuração.
NY3 - É uma revisão do NIS, usos de nomes hierárquicos e tables ao invés de mapas.

Existem dois tipos de servidores NIS:
Servidor master - O servidor master mantém os dados atualizados na rede e repassa aos clientes através de mapeamento slave e requisições dos clientes.
Servidor slave - O servidor slave mantém uma base de dados transferidos de um servidor master, este serviço é geralmente usado com redes mistas ou segmentado com vários hosts, pois mantém um tráfego menor de rede.

As principais ferramentas de configuração de clientes:
ypdomainname - Mostra o domínio do cliente NIS.
ypbind - Mostra o servidor de domínio NIS.
ypmake - Cria um mapeamento entre o cliente e o servidor NIS.
ypset - Relaciona um cliente NIS ao um servidor.
ypcat - Mostra a base de dados do servidor NIS.
ypwhich - Mostra o nome da máquina do servidor NIS.
yppoll - Mostra a versão do servidor NIS.
ypmatch - Faz uma procurar na base de dados do NIS.
yppasswd - Muda a senha do usuário na base de dados NIS.
ypchsh - Muda o shell da base de dados NIS.
ypchfn - Muda o nome do usuário da base de dados NIS.

Pacotes necessários para instalação:

yp-bind-x.x -> clientes samba
yptools-x.x -> servidor samba
ypserver -> ferramentas do samba

Configurando o NIS

Verifique se os arquivos /etc/gshadow e /etc/netgroup estão presentes, caso não crie os arquivos com o comando abaixo:

[root@localhost root]# touch /etc/gshadow

[root@localhost root]# touch /etc/netgroup

Utilize o aplicativo linuxconf para atribuir o domínio NIS:

[root@localhost root]# linuxconf

Ambiente de rede ->
NIS - sistema de informação de rede ->

Preencha os campos:

Domínio NIS [nome_do_domínio]
Servidor NIS [próprio_ip_da_máquina]

Selecione "Aceitar", "Sair", "Sair", "Ativar mudanças"

Marque o daemon responsável (ypserv e yppasswdd)

[root@localhost root]# ntsysv

[*] yppasswdd
[*] ypserv

Reinicie a máquina.

Após reiniciar a máquina verifique se o domínio está ativo digitando:

[root@localhost root]# nisdomainname

Basta agora adicionar os usuários localmente.

Toda vez que for adicionado ou excluído usuários, é necessário recriar os mapas NIS para realizar essa tarefa siga os passos abaixo:

Entre no diretório /var/yp

[root@localhost root]# cd /var/yp

Para recriar os mapas digite:

[root@localhost yp]# make

Desta forma o servidor está configurado.

Configurando o cliente

Ative o daemon responsável (ypbind):

[root@localhost root]# ntsysv

[*] ypbind

Digite o seguinte comando:

[root@localhost root]# authconfig

Em tipo de autenticação, marque a opção

(*) NIS

Adicione o nome do domínio:

Domínio NIS: [nome_do_domínio]
Pedido via anúncio [*]

Tente efetuar login como um usuário criado no servidor NIS.

Desta forma o cliente está devidamente configurado.

Firewall

Firewall é o mecanismo de segurança interposto entre a rede interna e a rede externa é com a finalidade de liberar ou bloquear o acesso de computadores remotos aos serviços que são oferecidos em um perímetro ou dentro da rede corporativa. Este mecanismo de segurança pode ser baseado em hardware, software ou uma mistura dos dois.

Três fatores estão em risco quando nos conectamos a Internet, são eles, a reputação, os computadores e as informações guardadas, e três fatores precisam ser resguardados, a privacidade, a integridade e a disponibilidade. Existem situações de riscos como, roubo de conexão depois dela ter sido autenticada, espionagem de dados secretos enquanto em trânsito pela rede e um usuário não autenticado convence a rede que ele foi autenticado.

Ele é o ponto de conexão com a Internet, tudo o que chega na rede interna deve passar pelo Firewall, ele é também o responsável por aplicar as regras de segurança, autenticar usuários, logar tráfego para auditoria e deve limitar a exposição dos hosts internos aos hosts da Internet, entretanto, algumas tarefas não podem ser executadas, como, proteger a rede contra usuários internos mal intencionados, conexões que não passam por ele, ameaças novas, no qual ele não foi parametrizado para executar uma ação.

Bastion Host

Bastion Host qualquer computador configurado para desempenhar algum papel crítico na segurança da rede interna, ele fica publicamente presente na Internet, provendo os serviços permitidos pela política de segurança da empresa.

Arquiteturas de Firewall

Normalmente, as empresas preferem implementar um Firewall baseado apenas em uma máquina, seja ele um host PC ou um roteador, entretanto, os Firewalls mais robustos, são compostos de várias partes.

Roteador com Triagem (Screening Router)

Essa é a maneira mais simples de se implementar um Firewall, pois o filtro, apesar de ser de difícil elaboraçao, é rápido de se implementar e seu custo zero, entretanto, se as regras do roteador forem quebradas, a rede da empresa ficará totalmente vulnerável.

Gateway de Base Dupla (Dual Homed Gateway)

Aqui, é posto uma única máquina com duas interfaces de rede entre as duas redes (a Internet e a rede da empresa). Quase sempre, esse Gateway, chamado de Bastion Host conta com um Proxy de circuito para autenticar o acesso da rede da empresa para a internet e filtrar o acesso da Internet contra a rede da empresa. Como na arquitetura anterior, se o Proxy for quebrado, a rede da empresa ficar totalmente vulnerável.

Iptables

O filtro de pacotes do kernel do Linux 2.4.X funciona por meio de regras estabelecidas na inicialização do sistema operacional. Todos os pacotes entram no kernel para serem analisados. As chains (correntes) são as situações possíveis dentro do kernel. Quando um pacote entra no firewall, o kernel verifica o destino dele e decide qual chain manipulará esse pacote. Isso é chamado de roteamento interno. Os tipos de chains irão depender da tabela que está sendo utilizada no momento. Existem três tabelas possíveis. O programa Iptables fornece uma interface para que o usuário possa manipular o filtro de pacotes do kernel.

Regras do iptables

A sintaxe do iptables é:

iptables [-t tabela]<comando> <chains> [opção<parâmetro>] <destino>

Tabelas

filter É a tabela-padrão, sendo usada quando nenhuma tabela for especificada. É usada quando há tráfego normal de dados, sem a ocorrência de NAT (Network Address Translation – Tradução de Endereços de Rede). Usa as chains INPUT, OUTPUT e FORWARD.

nat É utilizada quando há NAT. Exemplo: passagem de dados de uma rede privada para a Internet. Usa as chains PREROUTING, POSTROUTING e OUTPUT.

mangle É utilizada para efetuar alterações especiais em pacotes. Usa as chains PREROUTING e OUTPUT.

Comandos

Os comandos permitem que tarefas sejam executadas com o iptables. Os principais são:

-P -> Estabelece a observação de pacotes.
-A -> Adiciona uma regra (chave + opção + destino).
-D -> Apaga uma regra.
-F -> Apaga todas as regras.
-L -> Exibe o estado do iptables.
-h -> Exibe uma mensagem de ajuda.

Chains

As chains determinarão se a regra será aplicada quando um pacote tenta entrar, sair ou ser redirecionado pelo firewall.

INPUT **->** Verifica todos os pacotes que tentam entrar na rede interna.

OUTPUT -> Verifica todos os pacotes que tentam sair da rede interna.

FORWARD -> Verifica todos os pacotes que atravessam a rede, tanto da rede externa para a interna, como da rede interna para rede externa.

PREROUTING -> Analisa todos os pacotes que estão entrando no firewall para sofrerem NAT. O PREROUTING pode fazer ações de NAT com o endereço de destino do pacote. Isso é chamado de DNAT (Destination NAT).

POSTROUTING ->Analisa todos os pacotes que estão saindo do firewall para sofrerem NAT. O POSTROUTING pode realizar ações de NAT com o endereço de origem do pacote. Isso é chamado de SNAT (Source NAT).

Opções

-p -> Protocolo a ser verificado. Pode ser tcp, udp, icmp ou all. Também poe ser um valor ou nome retirado de /etc/protocols.

-s -> Dados da origem. São os dados da rede ou da máquina local. Pode ser -s<endereço IP></máscara de sub-rede>.

-d -> Dados do destino. São os dados da rede ou máquina de destino. Pode ser -d<endereço IP></máscara de sub-rede>.

-i -> Especifica a interface de entrada. Não pode ser utilizada com a chain OUTPUT.

-o -> Especifica a interface de saída. Não pode ser utilizada com a chain INPUT.

--sport -> Porta de origem. Só funciona com as opções -p udp e -p tcp.

--dport -> Porta de destino. Só funciona com as opções -p udp e -p tcp.

--syn -> Só se aplica em pacotes TCP com o bit SYN ligado e os bits ACK e FIN desligados. Tais pacotes são usados para requerer o início de uma conexão TCP; por exemplo, bloqueando-se tais pacotes vindos de uma interface, os pedidos de conexões TCP de entrada serão recusados, porém pedidos de conexões TCP de saída não serão afetados. Esta opção só tem sentindo quando o protocolo for TCP.

-j -> Determina o destino de uma regra.

Destinos

Por último, será determinado o que acontecerá com o pacote.

ACCEPT -> Permite a passagem do pacote pelo firewall.
REJECT -> Não permite a passagem do pacote pelo firewall e, no caso de pacotes ICMP, retorna uma mensagem host unreachable (computador inalcançável).
DROP -> Não permite a passagem do pacote pelo firewall e, no caso de pacotes ICMP, não retorna uma mensagem host unreachable (computador inalcançável).
LOG -> Cria um log referente à regra, em /var/log/messages. Usar antes de outras ações.
SNAT -> Utiliza-se com a chain POSTROUTING para fazer ações de mascaramento da origem.
DNAT -> Utiliza-se com a chains PREROUTING e OUTPUT para fazer ações de redirecionamento de portas e servidores, balanceamento de carga e proxy transparente. Caso a porta de destino não seja especificada, valerá a porta de origem. No firewall, a porta que será redirecionada não pode existir ou estar ocupada por um daemon.
--to -> Utiliza-se para definir o endereço IP e a porta de destino, após um DNAT ou de origem, após um SNAT.
MASQUERADE -> Faz mascaramento na saída de dados. Usado somente com a interface ppp0.
REDIRECT -> Redireciona uma requisição para uma porta local do firewall.
--to-port -> Define uma porta de destino, após um REDIRECT.
RETURN -> Executa regras até que haja uma falha em alguma delas. Utilizado em seqüências de linhas de firewall.

Extensões

As extensões permitem filtragens especiais, principalmente contra ataques de hackers. Elas podem ser usadas de duas maneiras: implicitamente, quando -p<protocolo> [extensão<opção>] é especificado ou quando opção -m <módulo <opção>> é especificada. Para sabermos quais extensões estão disponíveis para os protocolos, podemos utilizar o comando iptables-p<protocolo>-h; em que protocolo pode ser tcp, udp ou icmp. Para sabermos quais extensões estão disponíveis para os módulos, podemos utilizar o comando iptables -m<módulo>-h; em que módulo pode ser: mac, limit, multiport, mark, owner, state, tos e unclean.

Ativando o firewall

O Iptables Precisa ser carregado manualmente quando o kernel é recarregado.

Carregando os módulos principaais do iptables

#modprobe ip_tables
#modprobe iptable_filter
#modprobe ip_conntrack
#modprobe ip_conntrack_ftp

#modrpobe iptable_nat
#modprobe ip_nat_ftp

Agora temos que habilitar o encaminhamento IP, pois ele é desabilitado por padrão

 # vi /etc/sysctl.conf

net.ipv4.ip_forward = 1

ou

#echo 1 > /proc/sys/net/ipv4/ip_forward

Exemplos de Regras:

Lista as chains INPUT, FOWARD e OUTPUT.

#iptables -nL

Apaga qualquer regra existente e definindo nas politicas

#iptables -F
#iptables -t nat -F
#iptables -t mangle -F

Rejeita todos os pacotes

#iptables -P INPUT DROP
#iptables -P OUTPUT DROP
#iptables -P DROP

Rejeita entrada de icmp echo-reply no IP 192.168.1.10
#iptables -A INPUT -p 1 -icmp-type -d 192.168.1.10 -s 0/0 -j DROP

Habilita SNAT (MASCARAMENTO) em eth0
#iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

Servidor PROXY

Proxy e o responsável da ligação da rede interna com a rede externa permitindo uma melhor administração e segurança.

Vantagens de se utilizar um Proxy.

Cache -> o cache perimiti armazenar sites tornando-o mais rápido para maquinas da rede interna que acessam a internet.

Controle -> concerteza a melhor funcionalidade do proxy é o controle de acesso, onde podemos controlar quem pode acessar a internet e que sites estes não podem acessar.

SQUID

Squid é um proxy-cache de alta performance para clientes web, suportando protocolos FTP, gopher e HTTP.

O Squid mantém meta dados e especialmente objetos armazenados na RAM, cacheia buscas de DNS e implementa cache negativo de requests falhos.

Ele suporta SSL, listas de acesso complexas e logging completo. Por utilizar o Internet Cache Protocol, o Squid pode ser configurado para trabalhar de forma hierárquica ou mista para melhor aproveitamento da banda.

Podemos dizer que o Squid consiste em um programa principal - squid -, um sistema de busca e resolução de nomes - dnsserver - e alguns programas adicionais para reescrever requests, fazer autenticação e gerenciar ferramentas de clientes.

Podemos executar o Squid nas principais plataformas do mercado, como Linux, Unixes e Windows.

Pacotes necessários para instalação:

squid-x.x -> servidor proxy squid

Configurando o SQUID

Para uma configuração Básica no Squid pricisamos apenas descomentar algumas linhas no **squid.conf.**

Vamos aqui fazer a configuração básica do squid.

http_port 3128 -> Este parâmetro define a porta em que o serviço Squid irá escutar requisições, por padrão é 3128, mas você pode altera-la.

cache_mem 8M -> Este parâmetro define a quantidade de memória que o servidor Squid usará.

cache_dir ufs /var/spool/squid 100 16 256 -> Este parâmetro define o diretório onde o Squid alocará os arquivos para cache.

cache_access_log /var/log/squid/access.log -> Define o arquivo de log do Squid.

cache_mgr root@dominio.com -> Este parâmetro tem a finalidade de especificar o e-mail do

cache_effective_user squid -> Informa ao Squid com qual *nome de usuário* ele deve rodar

cache_effective_group squid -> Informa ao Squid com qual *grupo* ele deve rodar.

acl all src 0.0.0.0/0.0.0.0 -> Lista de controle de acesso

http_access alow all -> Libera o acesso a todos da acl de nome all

visible_hostname IP do servidor proxy -> Aqui você define qual o IP da máquina

Access Control Lists (Listas de Controle de Acesso)

As Listas de Controle de Acesso (Access Control Lists) ou simplesmente ACL's são os meios que o Squid nos dá para fazer uma filtragem e um melhor controle de permissões.

Nela podemos definir quem pode acessar ou não a internet, os sites que não podem e que podem serem acessados, enfim uma gama de funcionalidades de controle acesso.

Tipos de ACL's

src -> Endereço IP de origem, utilizado para restringir quais clientes podem fazer uso do proxy ou para identificar um host.

dst -> Endereço IP de destino, utilizado para restringir quais hosts remotos podem serem acessados ou para identificar um host remoto.

dstdomain -> Domínio de destino, utilizado para restrinfir acesso à um determinado domínio ou para identificar um domínio de destino.

time -> Hora e dia da semana, controla quando o proxy poderá ser utilizado e quando não poderá.

port -> Número da porta de destino, usado para restringir acesso a determinada porta de um servidor.

url_regex -> Utilizado para restringir determinadas URL's de acesso, a comparação de URL é baseada em expressão regular. Esse é um tipo de ACL que você irá utilizar muito.

proto -> Protocolo de transferência.

ident -> Nome de usuário.

proxy_auth-> Utilizado para requerer autenticação dos usuários e para especificar determinado suários dentro das ACL's.

Definindo ACL's

Dentro do arquivo de configuração do Squid, o squid.conf, você vai encontrar uma área que é a mais ideal para declarar as suas ACL's. Este espaço é onde as ACL's começam a ser definidas, facilmente identificada pela presença das mesmas.Para declarar ACL's, a sintaxe básica é a seguinte:

acl <nome da acl> <tipo da acl> <string> "<endereço de arquivo>"

acl porno url_regex -i sexo

http_access

A tag httpp_acces é utilizado em conjuto com as ACL's, digamos que lá você declarou tal arquivo, agora aqui no http_acces você irá decidir o que fazer com ele, no caso se vamos permitit ou se vamos negar.

Exemplos:

acl porno url_regex -i sexo
http_access deny porno

Aqui declaramos que a ACL de nome porno rejeitaria todos os sites que contenham a palavra sexo, em conjuto com o http_acces que nega todas as requisições com esta palavra.

acl net src 192.168.1.0/24
http_access allow net

Aqui criamos uma ACL de nome net com src, ou seja tudo que tem origem desta rede 192.168.1.0/24 poderá passar e utilizar a internet, porque permitimos no http_access.

Obs: Por padrão o Squid permiti que todos os usúarios acessem a internet, mas e de extrema importância que você comente a linhas onde estão **acl all src 0.0.0.0/0.0.0.0** e **http_access allow all**, e crie uma acl igual o exemplo acima permitindo apenas sua rede interna.

Bloqueando sites indevidos no proxy

Já foi dado um exemplo de como bloquear um site, mas e se quisermos bloquear vários sites de um só vez, sem que se necessite ficar criando diversas ACL's. O que faremos?

Criaremos um arquivo com as palavras ou sites bloqueados e defineremos apenas este arquivo na acl como no exemplo a seguir:

#vi /etc/squid/block

.parperfeito.com.br
.playboy
playboy.
sexy.com.br
.sexyclube.uol.com.br
.cracks.am
.bps.uol.com.br
.batepapo.uol.com.br
\.mpr3$
\.avi$

Aqui definimos quais sites e palavras serão bloqueados salve e sai do arquivo.

Dê permissão ao arquivo para que ele possa executar.

#chmod 755 /etc/squid/block

Agora vamos declara-la no arquivo de configuração do squid .

#vi /etc/squid.conf

Acrescente a acl's.

acl bloqueados url_regex -i "/etc/squid/block"

Agora bloqueie o acesso HTTP com o http_access:

http_access deny bloqueados

Proxy Transparente

A Proxy Transparente, nada mais é do que um redirecionamento para portas definidas no caso 3128, não necessitando a configuração manual de máquina a máquina com proxy.
Para habiltarmos o Proxy Transparente habiltamos as seguintes linhas no /etc/squid/squid.conf

#vi /etc/squid/squid.conf

httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on

Feito isso pricisamos habilitar o ip_forward e digitar um comando de redirecionamento no iptables da seguinte forma:

 # vi /etc/sysctl.conf

net.ipv4.ip_forward = 1

#iptables -t nat -A PREROUTING -s 192.168.1.0/24 -p tcp --dport 80 -j REDIRECT --to-port 3128

O arquivo de configuração do squid é muito extenso sendo disnecessário muitas linhas com comentários para isso vamos fazer uma cópia de segurança e a limpeza do arquivo.

 # cp /etc/squid/squid.conf /etc/squid/squid.conf.default
 # egrep -v "^#|^$" squid.conf.default > squid.conf

Enfim só pricisamos criar o diretório do cache e subir o serviço de daemon ou reler as configurações do squid da seguinte forma:

#squid -z -> cria o diretório do cache definido no squid.conf

#service squid stop

#service squid start
ou
#squid -k reconfigure

Caso esteja correto e não retonar nenhum erro vamos para o teste final, caso contrário reveja a linha em qual foi passado o erro.

Veja um exemplo de uma configuração de um squid.conf profissional

http_port 3128
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?

```
no_cache deny QUERY
cache_mem 8 MB
cache_swap_low 90
cache_swap_high 95
maximum_object_size 4096 KB
cache_dir ufs /var/spool/squid 100 16 256
cache_access_log /var/log/squid/access.log
ftp_user Squid@
#auth_param basic program /bin/ncsa_auth /etc/squid/squid_passwd
#auth_param basic children 5
#auth_param basic realm Digite o usuario e a senha
#auth_param basic credentialsttl 2 hours
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern .               0       20%     4320
#acl autenticacao proxy_auth REQUIRED
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 563
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443 563     # https, snews
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT
acl redeinterna src 192.168.1.0/24
acl palavra url_regex -i sex
acl bloqueados url_regex -i "/etc/squid/lists/blocked.conf"
acl porn dstdomain .arenadosexo.com .arquivosex.com .bol.com.br
#http_access allow autenticacao
http_access deny porn
http_access deny bloqueados
http_access deny palavra
http_access allow redeinterna
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost
http_reply_access allow all
http_access deny all
```

icp_access allow all
visible_hostname 192.168.1.100
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
coredump_dir /var/spool/squid

Testando o Squid

Abra um browser em uma máquina que esteje dentro da rede especificada no squid.conf e veja se consegue acessar os sites em que você bloqueou no arquivo do servidor proxy.

Documentações

Todas as documentações e arquivos referêntes a esta apostila estarão disponíveis no site:

www.raylinux.com