MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO TOCANTINS
CAMPUS PARAÍSO DO TOCANTINS

Diuliano de Sousa Oliveira
Eduardo Barros de Castro
Everton Borges da Silva
Kalid Rian Bucar

TUTORIAL FIREWALL E PROXY

PARAÍSO DO TOCANTINS – TO
Dezembro 2009

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO TOCANTINS
CAMPUS PARAÍSO DO TOCANTINS

Diuliano de Sousa Oliveira
Eduardo Barros de Castro
Everton Borges da Silva
Kalid Rian Bucar

TUTORIAL FIREWALL E PROXY

"Trabalho apresentado como requisito parcial de avaliação das disciplinas de Segurança de Sistema e Sistemas Operacionais do Curso Técnico Subseqüente em Informática, orientado pelo professor Ivo Sócrates M. de Oliveira".

PARAÍSO DO TOCANTINS – TO
Dezembro 2009

Esse tutorial vem com o intuito de tornar possível a implementação de um servidor firewall por meio do software Iptables, (lembrando que no Ubuntu 9.04 já possui este instalado por padrão) e de um Proxy por meio do Squid.

Lembrando que aqui será descrito as configurações apenas para o sistema operacional Ubuntu Desktop 9.04.

O QUE É FIREWALL?

Por intermédio de "portas", um computador realiza suas conexões, sejam elas locais ou remotas. Segundo Alan Guerreiro um Firewall é uma combinação de hardware e software cuja função é proteger, de intrusos indesejados, as redes dos computadores.

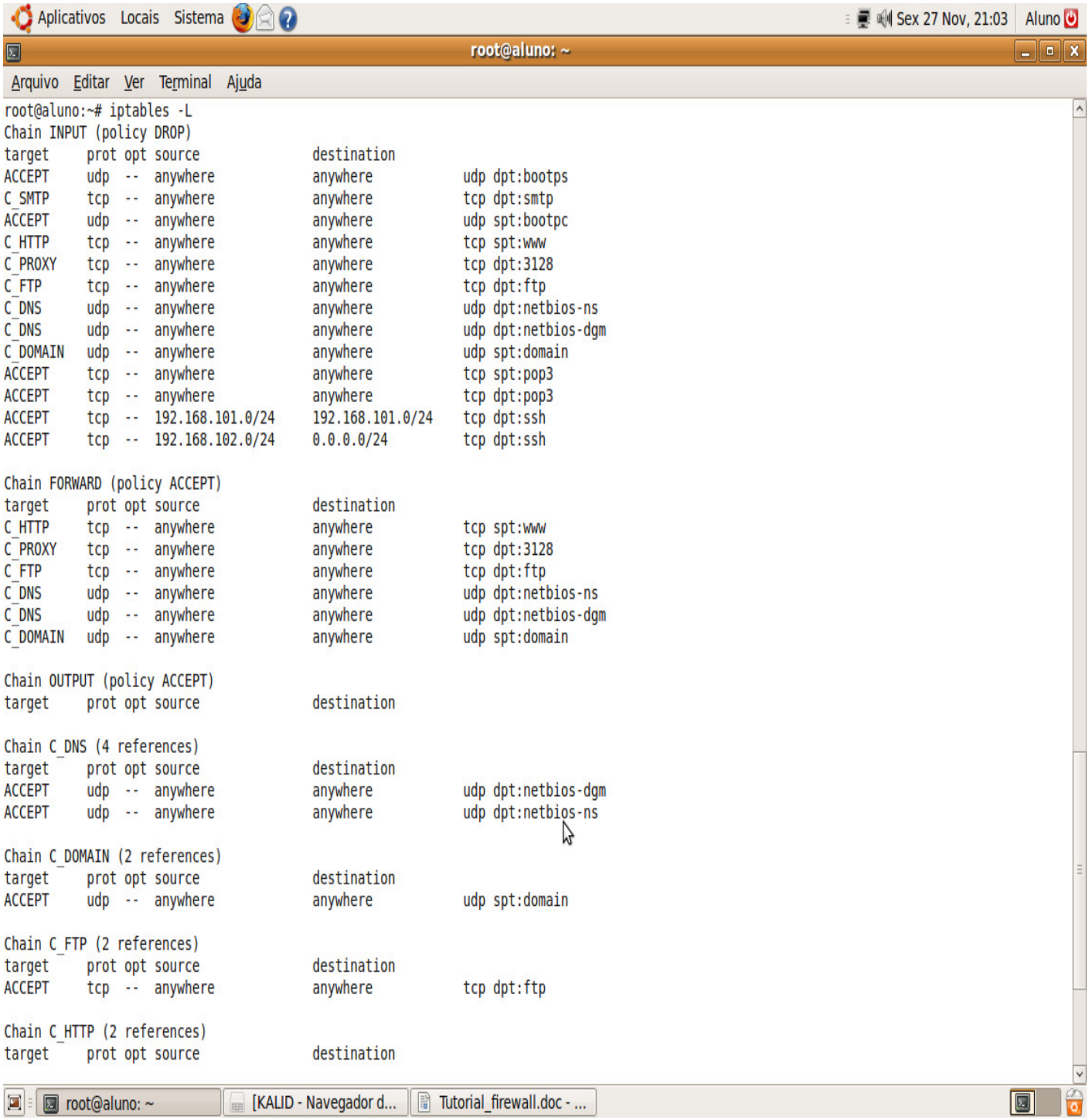
Muitas vezes as redes privadas criam firewalls para estarem seguros e que ninguém acesse os dados da empresa, a não serem aqueles que estão autorizados a isso.

Imaginamos que determinada empresa possui uma Intranet privada, e que apenas os seus funcionários a podem utilizar. A empresa poderá querer que estes funcionários tenham acesso à informação enquanto estão fora do local da empresa. Se o firewall funcionar corretamente, apenas os usuários autorizados podem acessar ao sistema e os usuários não autorizados terão o acesso interditado.

1° PASSO – LIMPANDO AS TABELAS DO IPTABLES

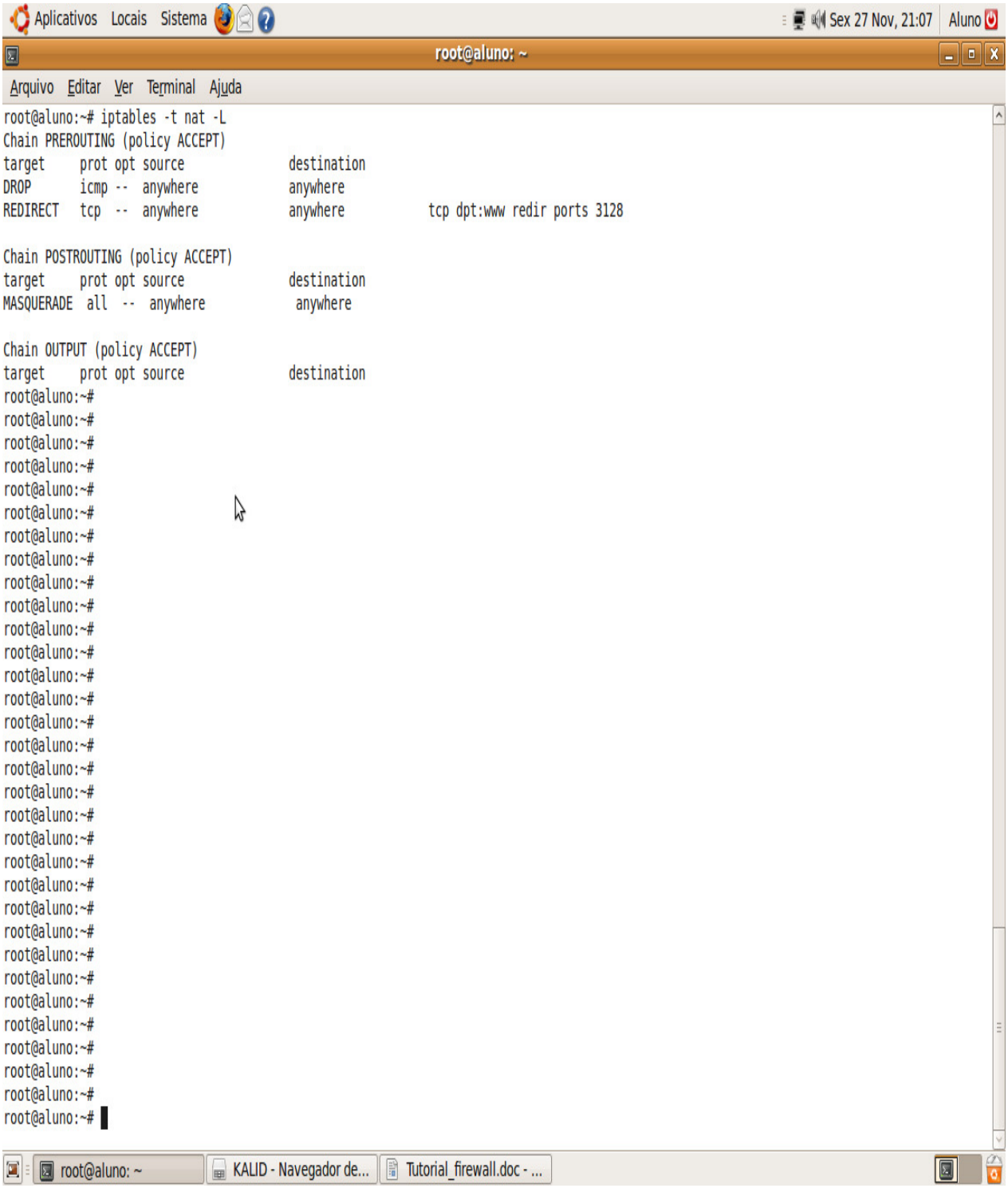
Com o Sistema Operacional instalado abra o terminal e autentique como root (para logar como root basta digitar o comando **su root** e logo em seguida digitar a senha e apertar enter) para que assim o mesmo tenha todas as permissões no Sistema Operacional. Podendo assim digitar as seguintes linhas de comando para listar as tabelas filter, nat e mangle e identificar o conteúdo existente.

iptables –L #Essa linha de comando esta listando o conteúdo da tabela filter.
Na imagem a seguir estamos listando as regras da tabela filter.

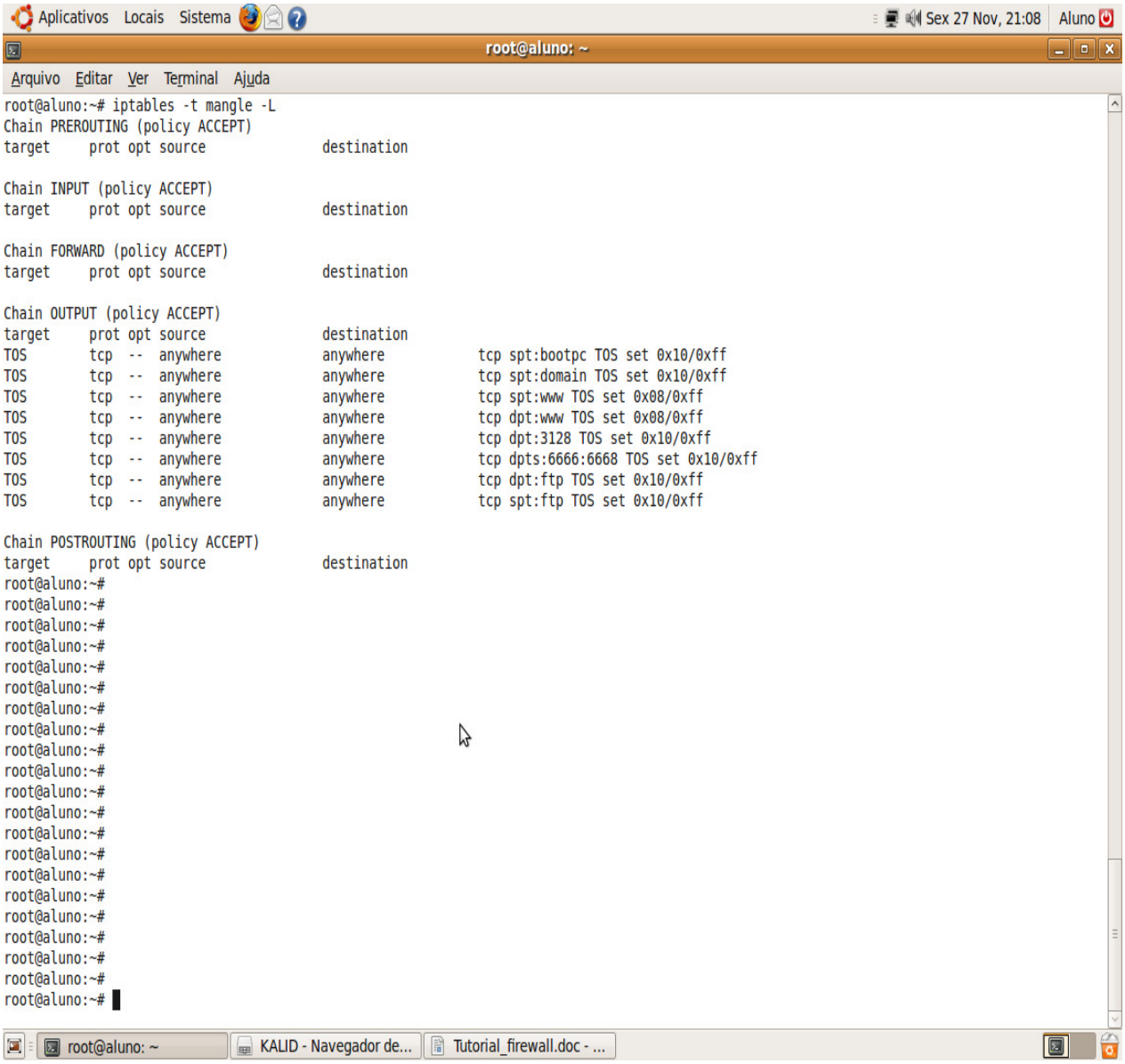


itpables –t nat –L #Essa linha de comando esta listando o conteúdo da tabela nat.

Na imagem a seguir estamos listando o conteúdo da tabela nat.



iptables –t mangle –L #Essa linha de comando esta listando o conteúdo da tabela mangle e a imagem a seguir esta dando um exemplo do conteúdo da tabela mangle.



Usando tais comandos o usuário deve verificar se todas as tabelas estão vazias, mas caso não estejam deve digitar os seguintes comandos para assim limpa-las.

iptables –F #Essa linha de comando esta apagando o conteúdo da tabela filter.

itpables –t nat –F #Essa linha de comando esta apagando o conteúdo da tabela nat.

iptables –t mangle –F #Essa linha de comando esta apagando o conteúdo da tabela mangle.

2º PASSO – HABILITANDO O FORWARD NO KERNEL DO LINUX

Para permitir que o Iptables faça forward de pacotes, essa opção deve ser ativada no kernel do sistema, forward é quando um pacote é direcionado ao servidor firewall, mas não é necessariamente para ele e sim para outro host dentro da rede, mas para que seja feita tal ação é preciso que se digite a seguinte linha de comando no terminal:

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

3° PASSO – CONFIGURAÇÃO BÁSICA DO IPTABLES

Agora serão mostrados alguns exemplos de configuração básica do Iptables, lembrando: cada configuração realizada ficará apenas na memória RAM, ou seja, se o computador for reiniciado perderá estas novas configurações. Para que a configuração fique armazenada no HD permanentemente utiliza-se seguinte linha de comando:

#Usando a linha de comando a seguir para se escolher o local onde será salvo as novas regradas editada pelo usuário:

iptables-save > /definindo o local que ira ser salvo

#Para restaurar as regras que o firewall seguia até as modificações que o usuário fez basta utilizar o comando a seguir:

iptables-restore < /local do arquivo

#Para que o sistema faça uma inicialização das regras automaticamente deve se localizar o seguinte arquivo:

nano /etc/rc.local

#E dentro do arquivo **nano /etc/rc.local** digitar a seguinte linha de comando para que assim as regras se iniciem automaticamente juntamente com a inicialização do sistema operacional.

iptables-restore < /local do arquivo

REGRAS DE FIREWALL PARA BLOQUEAR TODO O FLUXO DE DADOS, LIBERANDO APENAS O FLUXO NA INTERFACE ETH0

#
Definindo as regras padrão
#
iptables -P INPUT DROP #Definindo o padrão drop para tabela filter.
iptables -P OUTPUT DROP #Definindo o padrão drop para tabela filter.
iptables -P FORWARD DROP #Definindo o padrão drop para tabela filter.
#
Liberando a interface eth0
#
iptables -A INPUT -i eth0 -j ACCEPT
iptables -A OUTPUT -o eth0 -j ACCEPT

REGRA DE FIREWALL PARA LIBERAR O ACESSO NO SERVIDOR WEB AOS PROTOCOLOS HTTP E HTTPS

#
Definindo as regras padrão
#
iptables -P INPUT DROP #Definindo o padrão drop para tabela filter.

iptables -P OUTPUT ACCEPT #Definindo o padrão de liberação para tabela filter.

iptables -P FORWARD DROP #Definindo o padrão drop para tabela filter na cadeia Forward.
#
Definindo as regras de filtragem ao HTTP e HTTPS
#
iptables -A INPUT -p tcp –dport 80 -j ACCEPT #Liberando trafego na porta 80.
iptables -A INPUT -p tcp –dport 443 -j ACCEPT #Liberando trafego na porta 443.

FIREWALL PARA LIBERAR O ACESSO SERVIÇO HTTP E ACESSO PELA INTRANET NA PORTA 80

#
Definindo as regras padrão
#
iptables -P INPUT DROP # Definindo o padrão drop para tabela filter.
iptables -P OUTPUT ACCEPT #Definindo o padrão de liberação para tabela filter.

i**ptables -P FORWARD DROP** #Definindo o padrão drop para tabela filter na cadeia Forward.
#
definindo as regras de filtragem ao HTTP e HTTPS
#
iptables -A INPUT -p tcp –dport 80 -i eth1 -j ACCEPT #Liberando trafego na porta 80 na interface eth1.

iptables -A INPUT -p tcp –dport 443 -i eth0 -j ACCEPT #Liberando trafego na porta 443 na interface eth0.

REGRA DE FIREWALL PARA HOSPEDAR OS SERVIÇOS: HTTP, HTTPS, IMAP, POP3, SMTP, FTP, SSH

#
definindo as regras padrão
#
iptables -P INPUT DROP # Definindo o padrão drop para tabela filter.
iptables -P OUTPUT ACCEPT #Definindo o padrão de liberação para tabela filter.
iptables -P FORWARD DROP #Definindo o padrão drop para tabela filter na cadeia Forward.
#
liberando acesso pela internet eth0
#

iptables -A INPUT -p tcp –dport 80 -i eth0 -j ACCEPT #Liberando trafego na porta 80 na interface eth0.

iptables -A INPUT -p tcp –dport 443 -i eth0 -j ACCEPT #Liberando trafego na porta 443 na interface eth0.

#

liberando acesso pela internet (eth1)

#

iptables -A INPUT -p tcp –dport 80 -i eth1 -j ACCEPT #Liberando trafego na porta 80 na interface eth1

iptables -A INPUT -p tcp –dport 443 -i eth1 -j ACCEPT #Liberando trafego na porta 443 na interface eth1

iptables -A INPUT -p tcp –dport 143 -i eth1 -j ACCEPT #Liberando trafego na porta 143 na interface eth1

iptables -A INPUT -p tcp –dport 110 -i eth1 -j ACCEPT #Liberando trafego na porta 110 na interface eth1

iptables -A INPUT -p tcp –dport 25 -i eth1 -j ACCEPT #Liberando trafego na porta 25 na interface eth1

iptables -A INPUT -p tcp –dport 21 -i eth1 -j ACCEPT #Liberando trafego na porta 21 na interface eth1

iptables -A INPUT -p tcp –dport 22 -i eth1 -j ACCEPT #Liberando trafego na porta 22 na interface eth1

Lembrando que depois de feito as configurações desejadas pelo usuário se torna de suma importância que refaça o procedimento para armazenamento permanente no HD, assim como é ensinado na página sete (7) desse tutorial.

O QUE É UM PROXY?

Segundo Eduardo Campo, **proxy** é um servidor que atende a requisições repassando os dados do cliente a frente. Um usuário (cliente) conecta-se a um servidor proxy, requisitando algum serviço, como um arquivo, conexão, website, ou outro recurso disponível em outro servidor.

Um servidor proxy pode, opcionalmente, alterar a requisição do cliente ou a resposta do servidor e algumas vezes pode disponibilizar este recurso sem nem mesmo se conectar ao servidor especificado. Pode também atuar como um servidor que armazena dados em forma de cache em redes de computadores. São instalados em máquinas com ligações tipicamente superiores às dos clientes e com poder de armazenamento elevado.

INSTALANDO SQUID

O Squid é o aplicativo que nos será usado para que faça o trabalho de Proxy, lembrando que assim como o Iptablles ele também possui no banco de dados do sistema operacional que foi baseado esse tutorial (Ubuntu 9.04), mas é preciso que se digite (logado como root) os seguintes comandos para então fazer a instalação do mesmo.

apt-get update #Essa linha de comando esta baixando a atualização para o banco de dados do sistema operacional utilizado para que assim se baixe a ultima versão do Squid.

apt-get install squid #Após atualizado o banco de dados do sistema o usuário deve digitar essa linha de comando para então instalar a ultima versão do Squid.

Para que o proxy comece a funcionar o usuário deve digitar os seguintes comandos:

http_port 3128 transparen # Deixa o proxy funcionando transparentemente.

visible_hostname [nome de sua preferencia] #Linha de comando que dá o nome ao Proxy.

Acl all src 127.0.0.1 #Referenciando todos os IPs possíveis.

http_access allow all #Liberando o acesso a todos os IPs possíveis.

ACL - São regras que serão referenciadas junto ao “http_acess” podendo ser liberas ou negadas.
Para se especificar um padrão para a ACL, segue a seguinte ordem.

ACL [Nome da ACL] [tipo da ACL] [dado da ACL]

EXEMPLO DE CONFIGURAÇÃO DO SQUID

http_port 3128 transparent #Deixa o Proxy funcionando transparentemente.
visible_hostname Diulianoproxy #Linha de comando que da nome ao Proxy.

DEFININDO CACHE DE PÁGINAS DE ARQUIVOS

cache_mem 32 MB #Quantidade da memória que ira ser alocada para o cachê.

maximum_object_size_in_memory 64 KB #Linha que identifica o tamanho máximo dos arquivo que serão guardados no cachê.

maximum_object_size 512 MB #Linha que identifica o tamanho máximo dos arquivos que estarão em disco.

minimum_object_size 0 KB MB #Linha que identifica o tamanho mínimo dos arquivos que estarão em disco.

cache_swap_low 90 #Quantidade mínima de arquivo guardados em disco do cache do squid.

cache_swap_high 95 #Quando o cache do squid chegar a 95% o mesmo começa a descartar os arquivos mais antigos.

cache_dir ufs /var/spool/squid 2048 16 256 #Linha que criará um diretório para o armazenamento de informações de cache. O 2048 é o tamanho em mega byts o 16 é o número de pastas que serão criadas e 256 o números de subpastas em cada pasta.

cache_access_log /var/log/access.log #Arquivo que fará controle de log, o mesmo irá gerar log de todas as páginas acessadas.

auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/ #Linha de parâmetro para a autenticação.

auth_param basic realm Digite seu Login #Linha que pede a autenticação para o usuário.

acl all src 0.0.0.0/0.0.0.0 #Linha de comando que adiciona todos os IP's para essa regra.

acl manager proto cache_objeto #Referenciando o cache.

acl localhost src 127.0.0.1/255.255.255.255 #Criando regra para o local host.

acl redelocal src 192.168.101.0/24 #Criando regra para a rede local.

acl SSL_ports port 443 563 #Criando regra para portas 443 563.

acl Safe_ports port 80 #Criando regra para porta HTTP.

acl Safe_ports port 21 #Criando regra para porta FTP.

acl Safe_ports port 443 563 #Criando regra para porta HTTP.

acl Safe_ports port 70 #Criando regra para porta gopher.

acl Safe_ports port 210 #Criando regra para porta wais.

acl Safe_ports port 280 #Criando regra para porta http-mgmt.

acl Safe_ports port 488 #Criando regra para porta GSS-HTTP.

acl Safe_ports port 591 #Criando regra para porta filemaker.

acl Safe_ports port 777 #Criando regra para porta multiling HTTP.

acl Safe_ports port 901 #Criando regra para porta swat.

acl Safe_ports port 22 #Criando regra para porta SSH.

acl Safe_ports port 1025-65535 #Criando regra para porta portas altas.

acl diuliano src 192.168.101.251 #Criando regra para um IP específica.

acl purge method PURGE #Criando regra para método.

acl CONNECT method CONNECT #Criando regra para método.

acl blockGo url_regex /etc/squid/sitesblotxt #Regra que faz a leitura do arquivo onde contém os nomes dos sites acessados.

acl blockexten urlpath_regex .\exe$ #Regra que analisa um arquivo pela extensão.

http_access allow diuliano #Liberando regra que contem um endereço IP.

http_access deny blockGo #Negando regra que bloqueia sites.

http_access allow manager localhost #Liberando acesso ao local host.

http_access deny manager #Negando regra manager.

http_access allow purge localhost #Liberando acesso local host que passa pela regra purge.

http_access deny purge #Negando a regra purge.

http_access deny !Safe_ports #Todas as portas que não estiverem referenciadas nesta regra serão negadas.

http_access deny CONNECT !SSL_ports #Negando conexão as portas diferentes das que foram referenciadas.

http_access allow localhost #Liberando o acesso aos IPs do local host.

http_access allow redelocal #Liberando o acesso aos IPs da regra rede local.

http_access deny al #Negando os demais acessos.

Se o usuário seguiu as dicas aqui presentes nesse tutorial, então certamente o Firewall e o Proxy estarão basicamente configurados e rodando perfeitamente, fazendo bloqueios de varias portas, determinando o que pode ou não passar pela rede, entre várias outras regras aqui descritas.