Squid Proxy

Sumário

Índice de tabelas

Índice de Figuras

Capítulo 1

Squid Proxy

- Principais diretivas do Squid e comandos uteis;
- Bases de autenticação do Squid;
- Controle de banda;
- Squid-Graph.

1.1. Mãos a obra

O arquivo de configuração do Squid vem com quase 5000 linhas, devido aos comentários das diretivas que explicam o uso de cada uma. Dependendo do cenário de sua empresa uma configuração com um pouco mais de 40 linhas, já é suficiente para ter Squid bem configurado atendendo regras de bloqueios, otimização do cache e controle de banda. Vamos a prática:

Abra o arquivo de configuração do Squid para ver a descrição das principais diretivas.

```
# vim /etc/squid/squid.conf
```

Principais diretivas do Squid

http_port → Define a rede e porta em qual o Squid ira responder. Exemplo:

```
# Squid normally listens to port 3128
http_port 192.168.200.10:3128
```

visible_hostname → Exibe o nome do servidor que será apresentando nas mensagens de erro para os clientes. Exemplo:

```
#Default:
visible_hostname proxy.empresa.com.br
```

cache_dir ufs → Permite ajustar o cache em disco. É preciso definir na ordem o diretório no qual o Squid armazena os arquivos do cache, a quantidade de espaço no HD em MB que será usada para o cache, a quantidade de diretórios e a quantidade de subdiretórios. Exemplo:

```
#Default:
cache_dir ufs /var/spool/squid 512 128 256
```

cache_mem → Define o cache que será armazenado em memória. Exemplo:

```
#Default:
cache_mem 16 MB
```

maximum_object_size_in_memory → Define o tamanho em KB dos arquivos, que o cache vai armazenar na memória. Exemplo:

```
#Default:
maximum_object_size_in_memory 128 KB
```

minimum_object_size → Define o tamanho minimo para ficar armazenado em cache. Exemplo:

```
#Default:
minimum_object_size 0 KB
```

maximum_object_size → Define o tamanho máximo para ficar armazenado em cache. Exemplo:

```
#Default:
maximum_object_size 256 MB
```

cache_swap_low / cache_swap_high → Define que quando o cache atingir 95% de uso, serão descartados arquivos antigos até que a percentagem volte para um número abaixo de 90%. Exemplo:

```
#Default:
cache_swap_low 90
cache_swap_high 95
```

error_directory → Determina em qual linguagem serão apresentadas as mensagens de erro aos clientes. Exemplo:

```
#Default:
error_directory /usr/share/squid/errors/Portuguese
```

Comandos uteis no Squid

O Squid conta com uma serie de comandos para administração do serviço, vamos ver na prática a descrição de alguns deles.

Após configurar a diretiva cache_dir ufs, é necessário usar o comando abaixo para gerar os novos diretórios de cache. Antes de usar o comando pare o serviço.

```
# squid -z
```

Antes de inciar o serviço ou reler o arquivo de configuração, verifique se a sintaxe está correta:

```
# squid -k parse
```

É possível ir mais além com as opções do comando squid -k
squid -k {reconfigure|rotate|shutdown|interrupt|kill|debug|check}

reconfigure → Faz com o Squid releia o arquivo de configuração. Exemplo:

```
# squid -k reconfigure
```

rotate → Faz com que o squid de o rotate log. Exemplo:

```
# squid -k rotate
```

shutdown → Derruba o Squid, mas antes espera as conexões fecharem. Exemplo:

```
# squid -k shutdown
```

interrupt → Derruba o Squid sem esperar o fim das conexões. Exemplo:

```
# squid -k interrupt
```

kill → Derruba o Squid sem esperar as conexões fecharem nem os logs. Exemplo:

```
# squid -k kill
```

debug → Faz com que o Squid gere log de depuração máxima, ate que seja enviado de novo. Exemplo:

```
# squid -k debug
```

check → Verifica se existe uma cópia do Squid em execução. Exemplo:

```
# squid -k check
```

Bases de autenticação do Squid

Através do Squid podemos utilizar um recurso muito útil e interessante que é a autenticação de usuários. Podemos usar bases internas ou externas de dados, fazendo integração com diversos serviços Opensource e até proprietários.

Como posso saber em quais serviços o Squid pode autenticar?

Através das bibliotecas instaladas no diretório /usr/lib/squid, cada uma com uma função a integração em bases internas ou externas. Vamos a prática:

Liste os arquivos do diretório /usr/lib/squid

```
# ls /usr/lib/squid
```

```
digest_pw_auth  ldap_auth       ntlm_auth     squid_kerb_auth   unlinkd
diskd-daemon    logfile-daemon  pam_auth      squid_ldap_group  wbinfo_group.pl
getpwnam_auth   msnt_auth       smb_auth      squid_session     yp_auth
ip_user_check   ncsa_auth       smb_auth.sh   squid_unix_group
```

Vamos a descrição de alguns módulos de autenticação

ncsa_auth → Permite que o Squid autentique usuários usando como base, um arquivo de texto. É necessário instalar o pacote apache2-utils e criar os usuários com o comando htpasswd. Exemplo:

```
auth_param basic program /usr/lib/squid/ncsa_auth /etc/quid/passwd
auth_param basic children 5
auth_param basic realm Digite o nome de usuario e senha
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
```

pam_auth → Permite que o Squid autentique usuários usando como base o PAM. Não precisa instalar nenhum pacote adicional, já que que os nomes dos usuários são obtidos do arquivo /etc/passwd e as senhas de /etc/shadow. Exemplo:

```
auth_param basic program /usr/lib/squid/pam_auth
auth_param basic children 5
auth_param basic realm Digite o nome de usuario e senha
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
```

smb_auth → Permite que o Squid autentique os usuários em um servidor Samba configurado como PDC. É necessário instalar e configurar um servidor Samba com PDC. Exemplo:

```
auth_param basic program /usr/lib/squid/smb_auth -W 4LINUX -U 192.168.200.10
auth_param basic children 5
auth_param basic realm Digite o nome de usuario e senha
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
```

ldap_auth → Permite que o servidor autentique os usuários em um servidor LDAP. É necessário instalar e configurar um servidor LDAP. Exemplo:

```
auth_param basic program /usr/lib/squid/ldap_auth -b "dc=empresa,dc=com,dc=br"
-f "uid=%s" -h 192.168.200.10
auth_param basic children 5
auth_param basic realm Digite o nome de usuario e senha
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
```

msnt_auth → Permite que o servidor autentique os usuários em um servidor Active Directory da Microsoft. É necessário instalar e configurar um servidor Windows 2003 com AD. Exemplo:

```
auth_param basic program /usr/lib/squid/msnt_auth
auth_param basic children 5
auth_param basic realm Digite o nome de usuario e senha
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
```

mysqlt_auth → Permite que o servidor autentique os usuários em uma base do MYSQL. É necessário instalar e configurar um servidor MYSQL e criar um script PHP em /usr/lib/squid/. Exemplo:

```
auth_param basic program /usr/lib/squid/mysqlt_auth
auth_param basic children 5
auth_param basic realm Digite o nome de usuario e senha
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
```

Controle de banda no Squid

Um recurso muito utilizado no Squid é o controle de banda, feito atraves do chamado “delay pools” definido no arquivo /etc/squid.conf. Na prática este recurso limita a banda que cada usuário pode usar e a banda total que todos os usuários somados poderão usar simultaneamente. Vamos prática:

Abra o arquivo de configuração do Squid e adicione a configuração abaixo:

```
# vim  /etc/squid/squid.conf
```

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
acl rede src 192.168.200.0/24

delay_pools 1
delay_class 1 2
delay_parameters 1 262144/262144 10240/10240
delay_access 1 allow rede

http_access allow rede
http_access allow localhost
http_access deny all
```

Em nosso exemplo foi criado uma ACL de origem de nome “rede”, para a rede 192.168.200.0. Abaixo as diretivas para realizar o controle de banda.

Descrição das diretivas:

delay_pools → Define o número de “delay pools” que você vai possuir. Em nosso caso apenas uma;

delay_class → Define a classe de cada delay pool;

delay_parameters → Define os parâmetros para uma delay pool. O valor 262144/262144 indica quanto de nossa banda sera usado para o Squid, no caso 2MB.

O Valor 10240/10240 indica que os usuários poderão fazer downloas a 10KBps.
delay_access → Determina em qual delay pool uma requisição será aplicada

Para fazer a conta use a seguinte formula
link vezes 1024 vezes 1024 divido por 8
Exemplo:

```
# expr 2 \* 1024 \* 1024 / 8
```

O resultado do calculo é de 262144 bytes por segundo.

Nas regras do Squid sempre usamos bytes, por isso é necessário fazer a conversão, dividindo o valor em kbits por 8 e multiplicando por 1024 para ter o valor em bytes.

Veja na tabela o valor em bytes para uso em MB para o Squid
1MB -> 131072 bytes por segundos
2MB -> 262144 bytes por segundos
3MB -> 393216 bytes por segundos
4MB -> 524288 bytes por segundos
5MB -> 655360 bytes por segundos

Uma dica é não usar toda a banda para o Squid, sempre deixe um pouco para outros protocolos e serviços. Um exemplo de uso prático é quando uma empresa tem apenas um link de 3MB, deixando 2MB para o Proxy e 1MB para outros serviços.

Como faço para calcular a taxa de download?

Use a seguinte fórmula:
taxa_de_download vezes 8

```
# expr  10  \*  8
```

Pegue o resultado e use em uma outra formula:

1024 vezes resultado_da_outra_formula dividido por 8

```
# expr 1024 \* 80 / 8
```

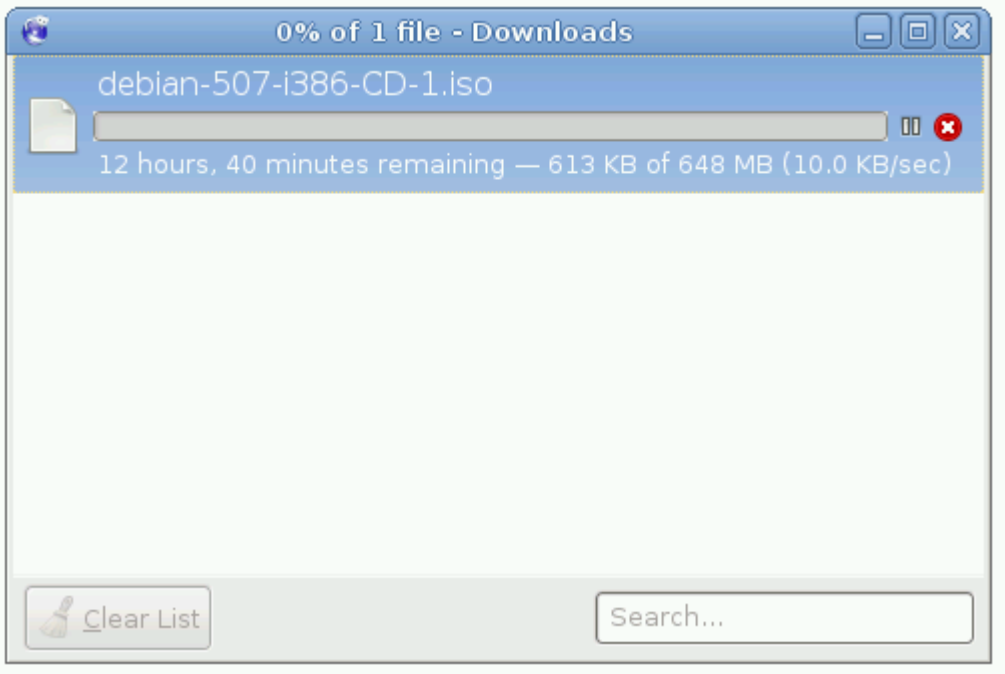
O resultado dos cálculos é de 10240. Abra o arquivo de configuração do Squid e na diretiva delay_parameters adicione o valor para 2MB e 10KBps. Exemplo:

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
acl rede src 192.168.200.0/24

delay_pools 1
delay_class 1 2
delay_parameters 1 262144/262144 10240/10240
delay_access 1 allow rede

http_access allow rede
http_access allow localhost
http_access deny all
```

Restarte o Squid e faça o teste no Browser fazendo download de algum arquivo

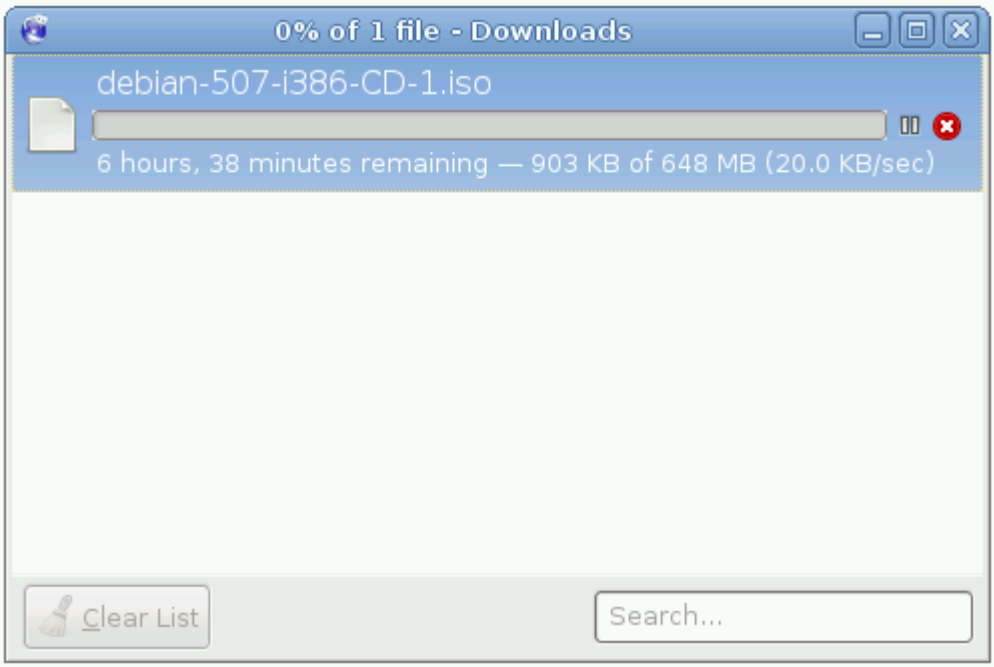


Exemplo de configuração para taxa de download a 20KBps

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
acl rede src 192.168.200.0/24

delay_pools 1
delay_class 1 2
delay_parameters 1 262144/262144 20480/20480
delay_access 1 allow rede

http_access allow rede
http_access allow localhost
http_access deny all
```



Squid-Graph

O Squid-Graph tem a função de gerar gráficos do Squid através dos logs. Este aplicativo que é escrito em Perl apresenta informações dos acessos e transferências de dados, usando como base o arquivo access.log. É necessário ter um servidor Web instalado como o Apache Vamos a prática:

Primeiro instale os pacotes “Perl” necessários para o Squid-Graph

```
# aptitude install libgd-barcode-perl libgd-gd2-noxpm-perl libgd-graph3d-perl libgd-graph-perl
```

Faça o download da versão estavel do aplicativo

```
# wget http://ufpr.dl.sourceforge.net/sourceforge/squid-graph/squid-graph-3.2.tar.gz
```

Descompacte para um diretório, seguindo a FHS vamos usar o /usr/local

```
# tar -xzvf squid-graph-3.2.tar.gz -C /usr/local
```

Crie um diretório em seu virtual host para o Squid-Graph

```
# mkdir /var/www/empresa.com.br/squid-graph
```

Gere os gráficos indicando a localização do script em perl, localização do diretório no virtual host, título dos gráficos e o log de acesso do Squid

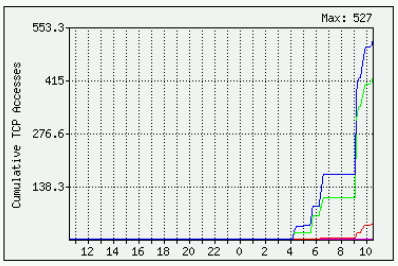
```
# /usr/local/squid-graph/squid-graph -c -n
-o=/var/www/empresa.com.br/squid-graph/ --title="Grafico de uso do Proxy" < /var/log/squid/access.log
```

Acesse os gráficos gerados em www.empresa.com.br/squid-graph

Grafico de uso do Proxy

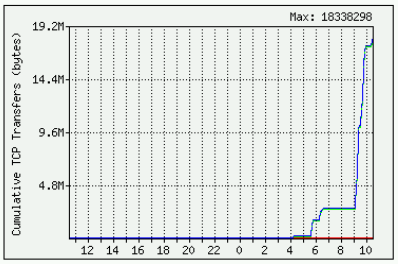
Generated:	Tue Dec 28 11:34:08 2010
Lines Analyzed:	527 lines (0 errors)
Analysis Duration:	1 seconds
Analysis Speed:	527 lines/sec
Graph Start:	Mon Dec 27 11:34:08 2010
Graph End:	Tue Dec 28 11:34:08 2010
Graph Domain:	24 hours (86400 seconds)

Cumulative graph of TCP Accesses



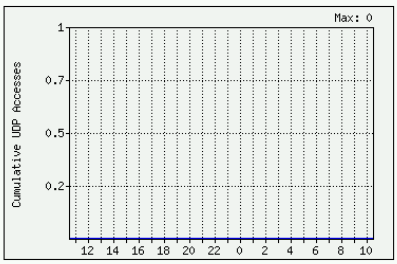
Total Accesses: 527
Average Accesses: 21.95 per hour
Total Cache Hits: 41
Average Cache Hits: 1.7 per hour
% Cache Hits: 7.77 %
Total Cache IMS Hits: 0
Average Cache IMS Hits: 0 per hour
Total Cache Misses: 429
Average Cache Misses: 17.87 per hour
% Cache Misses: 81.4 %

Cumulative graph of TCP Transfers



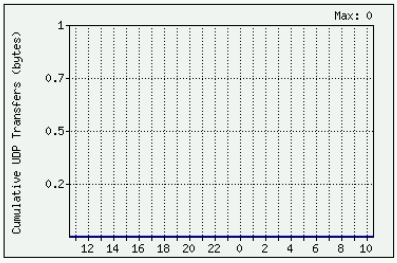
Total Transfers: 18.3 Mb
Average Transfers: 764 Kb per hour
Total Cache Hits: 61.4 Kb
Average Cache Hits: 2.5 Kb per hour
% Cache Hits: 0.33 %
Total Cache IMS Hits: 0 bytes
Average Cache IMS Hits: 0 bytes per hour
Total Cache Misses: 18.1 Mb
Average Cache Misses: 757.1 Kb per hour
% Cache Misses: 99.09 %

Cumulative graph of UDP Accesses



Total Accesses: 0
Average Accesses: 0 per hour
Total Cache Hits: 0
Average Cache Hits: 0 per hour
% Cache Hits: 0 %
Total Cache Misses: 0
Average Cache Misses: 0 per hour
% Cache Misses: 0 %

Cumulative graph of UDP Transfers



Total Transfers: 0 bytes
Average Transfers: 0 bytes per hour
Total Cache Hits: 0 bytes
Average Cache Hits: 0 bytes per hour
% Cache Hits: 0 %
Total Cache Misses: 0 bytes
Average Cache Misses: 0 bytes per hour
% Cache Misses: 0 %

Capítulo 2

Gerenciando

2.1. Objetivos

•Trobleshooting: Integração do Squid com Samba PDC e Windows 2003.

2.1. Troubleshooting

Como posso autenticar usuários em um Servidor Samba na rede?

Isso é possível através de um módulo de autenticação chamado smb_auth, que esta localizado no diretório /usr/lib/squid. Para que a autenticação funcione você precisa configurar no Squid:

A diretiva auth_param definindo o domínio do Samba e o IP do servidor;
Criar as ACLs do tipo origem e autenticação;
Alterar o script smb_auth.sh;

No Samba:
Instalar e configurar um servidor Samba como PDC;
Criar o compartilhamento Netlogon.
Criar o arquivo proxyauth no compartilhamento Netlogon;
Adicionar um usuário no Linux e no Samba.

Vamos a prática começando pelo Squid.

Abra o arquivo de configuração do Squid para configurar a diretiva auth_param e criar as ACLs.

```
# vim /etc/squid/squid.conf
```

```
auth_param basic program /usr/lib/squid/smb_auth -W 4LINUX -U 192.168.200.10
auth_param basic children 5
auth_param basic realm Digite o nome de usuario e senha
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
```

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
acl rede src 192.168.200.0/24
acl passwd proxy_auth REQUIRED

http_access allow redei passwd
http_access allow localhost
http_access deny all
```

Grave o arquivo e reinicie o serviço

```
# invoke-rc.d squid restart
```

Abra o script smb_auth.sh e faça uma alteração no arquivo, pq no Debian Lenny esse scriopt não consegue separar o nome do usuário da senha na hora de autenticar.

```
# vim /usr/lib/squid/smb_auth.sh
```

Faça a alteração de:

USER="$SMBUSER%$SMBPASS"
export USER

Para:

USER="$SMBUSER"

PASSWD="$SMBPASS"

export USER

export PASSWD

```
# Pass password to smbclient through environment. Not really safe.
USER="$SMBUSER"
PASSWD="$SMBPASS"
export USER
export PASSWD
```

No Samba configure como PDC e adicione o compartilhamento Netlogon

```
# vim /etc/samba/smb.conf
```

```
[netlogon]
        comment = Network Logon Service
        path = /var/lib/samba/netlogon
        guest ok = Yes
        share modes = No
```

Crie o diretório do compartilhamento conforme configurado

```
# mkdir /var/lib/samba/netlogon
```

Crie o arquivo proxyauth com o conteúdo “allow”, pois é este arquivo o que smb_auth ira procurar e ler para dar acesso ao usuário e senha cadastrado no Samba

```
# vim /var/lib/samba/netlogon/proxyauth
allow
```

Crie um usuário no Linux

```
# useradd -m -c “Tux Linux” -s /bin/false -N tux
```

E depois adicione no Samba

```
# smbpasswd -a tux
```

```
New SMB password:
Retype new SMB password:
Added user tux.
```

Para testar a autenticação use o comando smb_auth informando o dominio do Samba e o IP da maquina.

```
# /usr/lib/squid/smb_auth -W 4LINUX -U 192.168.200.10 -d
```

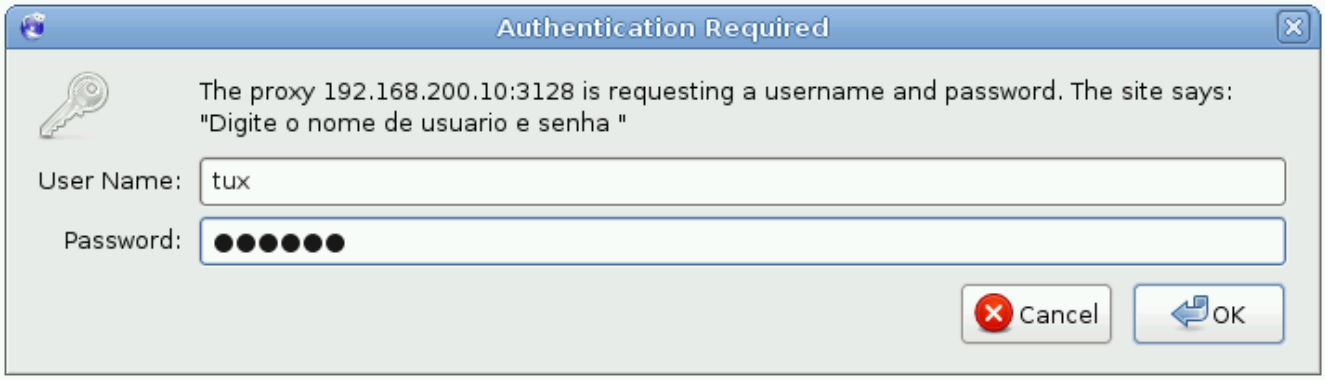
```
tux 123456
Domain name: 4LINUX
Pass-through authentication: no
Query address options: -U 192.168.200.10 -R
Domain controller IP address: 192.168.200.10
Domain controller NETBIOS name: SERVER
Contents of //SERVER/NETLOGON/proxyauth: allow
OK
```

-W → Define o domínio do PDC;

-U → IP do servidor Samba;

-d → Modo debug para verificar possíveis erros.

Para terminar, acesse o Browser e verifique se o usuário cadastrado no Samba realiza a autenticação no Squid.



Como posso autenticar usuários em um Servidor Active Directory na rede?

Isso é possível através de um módulo de autenticação chamado msnt_auth, que esta localizado no diretório /usr/lib/squid. Para que a autenticação funcione você precisa configurar no Squid:

A diretiva auth_param para usar o módulo msnt_auth;
Criar as ACLs do tipo origem e autenticação;
Criar o arquivo msntauth.conf definindo informações do servidor;
Criar os arquivos para bloquear e liberar usuários.

No Windows 2003:
Instalar e configurar um servidor Active Directory;
Adicionar um usuário no Active Directory.

Vamos a prática começando pelo Squid.

Abra o arquivo de configuração do Squid para configurar a diretiva auth_param e criar as ACLs.

 # vim /etc/squid/squid.conf

```
auth_param basic program /usr/lib/squid/msnt_auth
auth_param basic children 5
auth_param basic realm Digite o nome de usuario e senha
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
```

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
acl rede src 192.168.200.0/24
acl passwd proxy_auth REQUIRED

http_access allow redei passwd
http_access allow localhost
http_access deny all
```

Grave o arquivo e reinicie o serviço

```
# invoke-rc.d squid restart
```

Crie o arquivo de configuração que o módulo msnt_auth ira ler, contendo o nome do servidor PDC, BDC e o domínio do Windows 2003. Adicione também o caminho e nome dos arquivos para bloquear e liberar usuários.

```
# vim /etc/squid/msntauth.conf
```

```
server servidor servidor ad2003
denyusers /etc/squid/etc/denyusers
allowusers /etc/squid/etc/allowusers
```

Edite o arquivo /etc/hosts e informe o IP, FQDN e Alias do servidor Windows

```
# vim /etc/hosts
```

```
127.0.0.1       localhost.localdomain localhost
192.168.200.10  server.empresa.com.br   server
192.168.200.100 servidor.ad2003.com.br  servidor
```

Crie o diretório para armazenar arquivos de configuração

```
# mkdir /etc/squid/etc
```

Crie o arquivo denyusers e adicione os usuários que serão bloqueados no Proxy

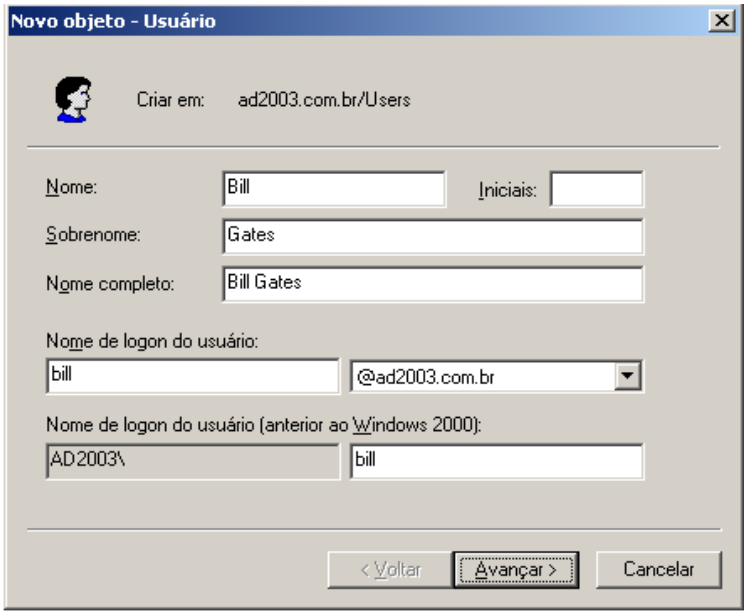
```
# vim /etc/squid/etc/denyusers
bill
steven
```

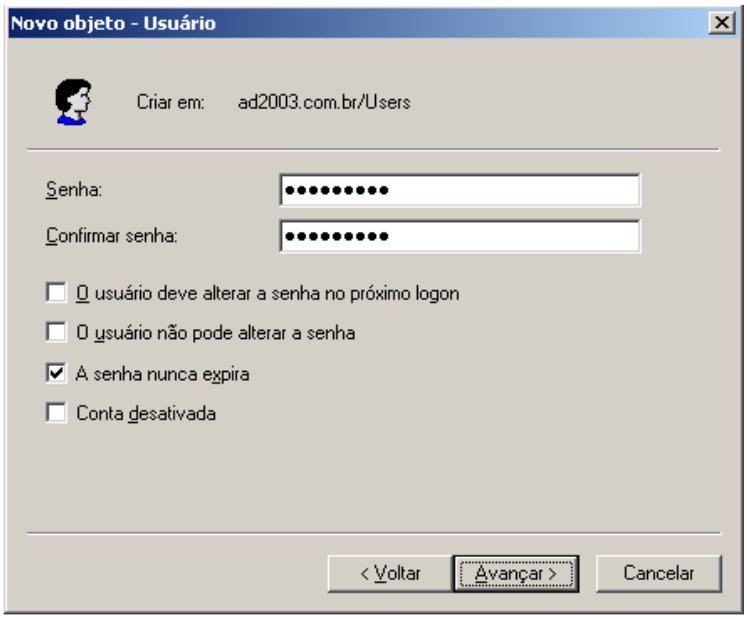
Crie o arquivo allowusers e adicione os usuários que serão liberados no Proxy

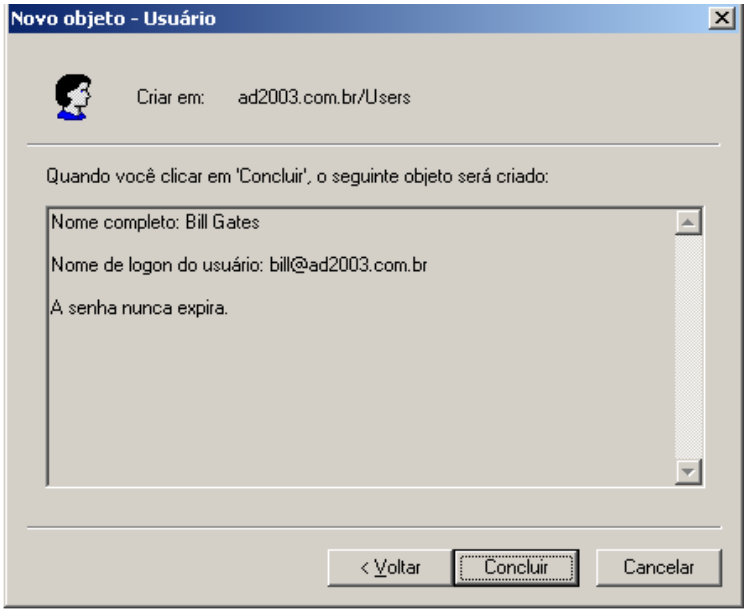
```
# vim /etc/squid/etc/allowusers
linus
```

No Windows 2003 configure o serviço Active Diretory, e adicione os usuarios clicando em Iniciar → Ferramentas administrativas → Usuários e computadores do Active Diretory.

Clique com o botão direito em Users → Novo → Usuário







Para testar a autenticação use o comando msnt_auth, e após teclar enter informe o nome de usuário e senha criado no Active Directory.

```
# /usr/lib/squid/msnt_auth
```

```
bill LInux2011
ERR
```

```
# /usr/lib/squid/msnt_auth
```

```
linus LInux2011
OK
```

Em nosso exemplo o resultado ERR foi exibido devido ao usuário bill, presente no arquivo denyusers e OK devido ao usuário linus no arquivo allousers.

Para terminar, acesse o Browser e verifique se o usuário cadastrado no Active Diretory realiza a autenticação no Squid.