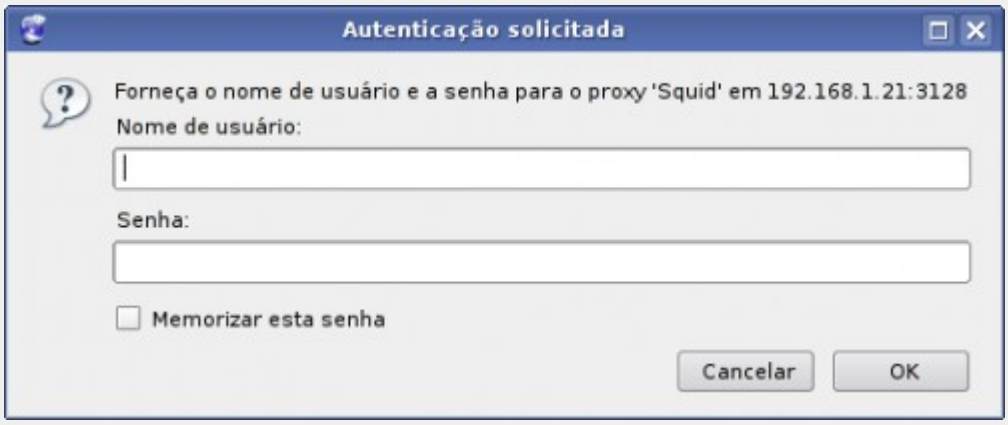
Prática dirigida

Proxy com autenticação

Você pode adicionar uma camada extra de segurança exigindo autenticação no proxy. Este recurso pode ser usado para controlar quem tem acesso à internet e auditar os acessos em caso de necessidade. Quase todos os navegadores oferecem a opção de salvar a senha, de modo que o usuário precisa digitá-la apenas uma vez a cada sessão:



A forma mais simples de implementar autenticação no Squid é usando o módulo "ncsa_auth", que faz parte do pacote principal. Ele utiliza um sistema simples, baseado em um arquivo de senhas, onde você pode cadastrar e bloquear os usuários rapidamente.

Para criar o arquivo de senhas, precisamos do script "**htpasswd**". Nas distribuições derivadas do Debian ele faz parte do pacote **apache2-utils**, que você pode instalar via apt-get:
apt-get install apache2-utils

No CentOS e no Fedora ele faz parte do pacote principal do apache (o pacote "httpd"), que pode ser instalado através do yum.

Em seguida, crie o arquivo que será usado para armazenar as senhas, usando o comando "touch" (que simplesmente cria um arquivo de texto vazio):

touch /etc/squid/squid_passwd

O próximo passo é cadastrar os logins usando o htpasswd, especificando o arquivo que acabou de criar e o login que será cadastrado, como em:

htpasswd /etc/squid/squid_passwd gdh

Depois de terminar de cadastrar os usuários, adicione as linhas que ativam a autenticação no squid.conf:

auth_param basic realm **Squid**
auth_param basic program **/usr/lib/squid/ncsa_auth /etc/squid/squid_passwd**
acl autenticados proxy_auth REQUIRED
http_access allow autenticados

O "auth_param basic realm Squid" indica o nome do servidor, da forma como ele aparecerá na janela de autenticação dos clientes; esta é na verdade uma opção meramente estética. O "/usr/lib/squid/ncsa_auth" é a localização da biblioteca responsável pela autenticação. Eventualmente, ela pode estar em uma pasta diferente dentro da distribuição que estiver usando; nesse caso, use o comando "**locate**" ou a busca do sistema para encontrar o arquivo e altere a linha indicando a localização correta. Finalmente, o "/etc/squid/squid_passwd" indica a localização do arquivo de senhas que criamos no passo anterior.

Estas quatro linhas criam uma acl chamada "autenticados" (poderia ser outro nome), que contém os usuários que se autenticarem usando um login válido. Estas linhas devem ser colocadas antes de qualquer outra regra que libere o acesso, já que, se o acesso é aceito por uma regra anterior, ele não passa pela regra que exige autenticação.

Entretanto, se você usar uma configuração similar a essa:

```
auth_param basic realm Squid
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/squid_passwd
acl autenticados proxy_auth REQUIRED
http_access allow autenticados

acl redelocal src 192.168.1.0/24
http_access allow localhost
http_access allow redelocal
http_access deny all
```

... vai notar que a regra de autenticação essencialmente desativa a regra que bloqueia o acesso de usuários fora da rede local. Todos os usuários tem acesso ao prompt de autenticação e todos que se autenticam ganham acesso, mesmo que estejam utilizando endereços fora da faixa usada na rede. Para evitar isso, é necessário restringir o acesso de usuários fora da rede local antes da regra de autenticação. Veja um exemplo:

```
# Bloqueia acessos de fora da rede local antes de passar pela autenticação:
acl redelocal src 192.168.1.0/24
http_access deny !redelocal
# Outras regras de restrição vão aqui, de forma que o acesso seja negado
# antes mesmo de passar pela autenticação:
acl bloqueados url_regex -i "/etc/squid/bloqueados"
http_access deny bloqueados

# Autentica o usuário:
auth_param basic realm Squid
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/squid_passwd
acl autenticados proxy_auth REQUIRED
http_access allow autenticados

# Libera o acesso da rede local e do localhost para os autenticados,
# bloqueia os demais:
http_access allow localhost
http_access allow redelocal
http_access deny all
```

Veja que agora usamos a regra "http_access deny !redelocal" no início da cadeia. A exclamação inverte a lógica da regra, fazendo com que ela bloqueie todos os endereços que não fizerem parte da acl "redelocal".

Ao implementar a autenticação, você passa a poder criar regras de acesso com base nos logins dos usuários e não mais apenas com base nos endereços IP. Imagine, por exemplo, que você queira que apenas dois usuários da rede tenham acesso irrestrito ao proxy. Os demais (mesmo depois de autenticados), poderão acessar apenas no horário do almoço, e quem não tiver login e senha válidos não acessa em horário nenhum. Neste caso, você poderia usar esta configuração:

```
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/squid_passwd
acl autenticados proxy_auth REQUIRED

acl permitidos proxy_auth gdh tux
acl almoco time 12:00-13:00

http_access allow permitidos
http_access allow autenticados almoco
```

Aqui temos os usuários que passaram pela autenticação divididos em duas regras. A acl "autenticados" inclui todos os usuários, enquanto a acl "permitidos" contém apenas os usuários gdh e tux.

Graças à regra "http_access allow permitidos", os dois podem acessar em qualquer horário, enquanto os demais caem na regra "http_access allow autenticados almoco", que cruza o conteúdo das acls "autenticados" e "almoço", permitindo que eles acessem, mas apenas das 12:00 às 13:00.

Além do módulo ncsa_auth que, como vimos, permite usar um arquivo de senhas separado, válido apenas para o proxy, o Squid suporta também um conjunto de módulos que permitem fazer com que o Squid se autentique em um servidor externo, integrando o proxy a um sistema de autenticação já existente.

O mais simples é o módulo **smb_auth**, que permite que o Squid autentique os usuários em um servidor Samba, configurado como PDC. Com isso, os usuários passam a utilizar o mesmo login e senha que utilizam para fazer logon. Para usar o smb_auth, você usaria a configuração a seguir, especificando o domínio (na opção -W) e o endereço do servidor PDC (na opção -U):

```
auth_param basic realm Squid
authenticate_ip_ttl 5 minutes
auth_param basic program /usr/lib/squid/smb_auth -W dominio -U 192.168.1.254
acl autenticados proxy_auth REQUIRED
http_access allow autenticados
```

Com o módulo smb_auth, o Squid simplesmente repassa o login e senha fornecido pelo usuário ao servidor PDC e autoriza o acesso caso o PDC retorne uma resposta positiva. Não é necessário que o servidor com o Squid faça parte do domínio, nem que exista uma cópia do Samba rodando localmente (são necessários apenas os pacotes "samba-client" e "samba-common", que correspondem ao cliente Samba), por isso a configuração é bastante simples. Naturalmente, para utilizar esta configuração você deve ter um servidor PDC na sua rede, como aprenderemos a configurar em detalhes no capítulo sobre o Samba.

Outros módulos que podem ser usados são o "squid_ldap_auth", que permite que o servidor autentique os usuários em um servidor LDAP e o "ntlm_auth", que permite integrar o servidor Squid ao Active Directory.