SEG12 - Semana 1 - Sessão 13

Francisco Marcelo, Marcelo Karam e Felipe Scarel

06-08-2018

Proxy Squid

Nesta sessão iremos instalar e configurar o Squid, uma solução de *proxy* web que provê funcionalidades de *cache* e redirecionamento. O Squid pode ser utilizado para diversos fins: acelerar o acesso web a partir da realização de *cache* de páginas acessadas com frequência, realizar *cache* de requisições web, DNS outros tipos de consulta para um grupo de usuários, e filtragem de acesso por domínio, URL e análise de conteúdo de páginas. Normalmente configura-se o Squid para trabalhar com os protocolos HTTP e FTP, mas também é possível filtrar requisições HTTPS através de inspeção SSL/TLS.

1) Instalação e configuração inicial do servidor *proxy* Squid

Esta configuração será realizada na máquina virtual *Server_Linux*.

Instale e configure o servidor *proxy* Squid na máquina *Server_Linux*, pacotes `squid3` e `sarg`. Configurações:

- Autorizar conexões vindas de ambas as redes internas, 192.168.0.0/24 e 172.16.0.0/24.
- Recusar demais conexões.
- Diretório de *cache* de páginas em `/var/spool/squid3`
- Log de acessos em `/var/log/squid3/access.log`
- Log geral do *proxy* em `/var/log/squid3/cache.log`
- Porta de acesso 3128/TCP.

2) Configuração do navegador cliente do *proxy*

Esta configuração será realizada na máquina virtual *Win7-padrao*.

Vamos testar a configuração realizada. Acesse a máquina *Win7-padrao* e configure o *proxy* do sistema para o IP da máquina *Server_Linux*. A seguir, acesse um website na porta 80/HTTP (sugestão: http://www.openbsd.org), teste se houve sucesso na conexão, e verifique se o log de acessos do Squid fez o *cache* das páginas solicitadas pelo usuário.

3) Configuração de controles de acesso

Esta configuração será realizada nas máquinas virtuais *Server_Linux* e *Win7-padrao*.

Vamos agora implementar controles de acesso ao servidor *proxy* usando ACLs (*Access Control Lists*). Para testar as configurações, evite usar websites HTTPS, pois o Squid está configurado para HTTP apenas; além disso, o navegador Internet Explorer da máquina *Win7-padrao* está bastante desatualizado. O website http://www.openbsd.org é um bom alvo para testes.

Implemente os seguintes controles:

a. Bloqueio via endereço físico (MAC) — `acl` com palavra-chave `arp`.

b. Bloqueio via endereço IP de origem — `acl` com palavra-chave `src`.

c. Bloqueio pela hora de acesso — `acl` com palavra-chave `time`. Utilize os comandos `date -s` e `hwclock --systohc` para ajustar o relógio do servidor para um horário proibido e testar sua configuração.

d. Bloqueio por expressão regular de extensão de arquivo — `acl` com palavra-chave `urlpath_regex`. Faça com que o acesso a qualquer arquivo com as extensões `.avi`, `.mp3` ou `.pdf` seja bloqueado. Use a pesquisa `site:ftp.openbsd.org filetype:pdf` no Google para encontrar um arquivo que se encaixe no bloqueio configurado.

e. Bloqueio por expressão regular de palavra em URL — `acl` com palavra-chave `urlpath_regex`. Faça com que qualquer URL que contenha as palavras `crypto`, `playboy`, `sexo`, `torrent` e `virus` seja bloqueada. Acesse a URL http://www.openbsd.org/crypto.html para testar a configuração.

f. Bloqueio por domínio de destino — `acl` com palavra-chave `dstdomain`. Faça com que qualquer acesso aos domínios `facebook.com`, `instagram.com`, `twitter.com` e `whatsapp.com` seja negado. Acesse a URL http://web.whatsapp.com para testar sua configuração.

3) Configuração do SARG

Esta configuração será realizada na máquina virtual *Server_Linux*.

Vamos agora configurar o *Squid Analysis Report Generator*, ou simplesmente SARG. O SARG é um gerador de relatórios de acesso do Squid, que analisa os arquivos de log deste para produzir informações relevantes para o administrador de sistemas.

Já instalamos o pacote do SARG na atividade 1 desta sessão. Configure-o da seguinte forma:

- Analisar log do Squid em `/var/log/squid3/access.log`.
- Produzir relatórios no diretório `/var/www/meusite/squid-reports`.
- Não resolver endereços IP para nomes.
- Usar formato de data no padrão europeu (mesmo utilizado no Brasil).
- Produzir relatórios no *charset* UTF-8.

Uma vez configurado o programa, rode o comando `sarg` como root e acesse a URL `https://meusite.empresa.com.br/squid-reports/` para visualizar os resultados.

4) *Proxy* transparente

Esta configuração será realizada nas máquinas virtuais *Server_Linux* e *Win7-padrao*.

Pode não ser interessante ter que configurar cada estação cliente para que utilize expressamente o *proxy*. É possível configurar o firewall da rede para redirecionar conexões às portas 80/HTTP e

443/HTTPS de forma automática para o *proxy*, sem editar as configurações de qualquer cliente — esse tipo de cenário é denominado *proxy* transparente.

Edite o firewall `iptables` da máquina *Server_Linux* para que os pacotes passantes com destino à porta 80/HTTP de um servidor externo sejam redirecionados para o Squid local, operando na porta 3128/TCP.

Use o pacote `iptables-persistent` para tornar suas configurações permanentes mesmo após o `reboot` da máquina. Na instalação do pacote, quando perguntado, responda:

Tabela 1. Configurações do iptables-persistent

Pergunta	**Resposta**
Salvar as regras IPv4 atuais?	Sim
Salvar as regras IPv6 atuais?	Sim

Não se esqueça de configurar o Squid em modo transparente. Finalmente, limpe as configurações de *proxy* da máquina *Win7-padrao*, e verifique que a *cache* e bloqueios do Squid permanecem operacionais.

Observe que todas as configurações desta sessão foram feitas para um *proxy* HTTP apenas. Embora funcional, muito sites hoje em dia utilizam HTTPS exclusivamente, o que torna nossa implantação apenas parcialmente útil.

O módulo *Peek and Splice* do Squid (https://wiki.squid-cache.org/Features/SslPeekAndSplice), disponível a partir da versão 3.5, permite a configuração de *proxy* para o protocolo HTTPS. O Squid, nesse caso, atua como uma espécie de *man-in-the-middle* entre a máquina cliente e o servidor remoto, forjando certificados para manter duas conexões criptografadas simultaneamente:

```
Cliente ⇐ ⇒ Squid ⇐ ⇒ Servidor Remoto
```

Assim, os dados passam em claro por dentro do próprio *proxy*.

A configuração desse módulo extrapola o escopo desta sessão, mas deixamos aqui nossa recomendação do mesmo para leitura futura.