FIAP ON

BIOHACKING,
DEEP WEB E CRIPTOGRAFIA

PROXY SSL/TLS

ALESSANDRO VINÍCIUS VIEIRA

8

LISTA DE FIGURAS

SUMÁRIO

8 PROXY SSL/TLS

O termo “privacidade” tem se tornado cada vez mais comum na internet, principalmente em publicações que falam sobre a falta da privacidade na Rede Mundial de Computadores. No entanto, quanto mais a internet caminha para uma total descortinação dos dados privados de qualquer cidadão ou instituição, mais as comunidades hacktivistas têm disseminado certas técnicas (muitas vezes nem tão modernas) para se prevenir de invasões de privacidade *on-line*, como, por exemplo, o uso de servidores Proxy para anonimizar conexões.

É lógico que o objetivo dos servidores Proxy vai muito além do que anonimização, como abordaremos neste capítulo, porém o assunto “proxy” tem ganhado relevância no cenário atual da internet, principalmente porque mais e mais usuários têm se preocupado com a exposição dos seus dados *on-line*.

8.1 O que é um Proxy?

Em sua grande maioria, os navegadores de páginas web fazem conexões diretas com a Internet, mas há outra forma bem mais interessante de conexão: eles podem ser configurados para se conectarem por meio de um servidor Proxy.

O Proxy é um serviço disponível em um ambiente servidor, que recebe requisições das estações de trabalho para conexões à internet, e seu papel fundamental é buscar a informação primeiramente no seu cache local e caso não encontre o documento requisitado, fazer a busca no site solicitado pela estação de trabalho. Na segunda situação, o endereço de Internet que fica registrado no servidor da página solicitada, é o do servidor Proxy, pois ele é o dispositivo que está entre a rede local e a internet.

Conforme Resende e Stella (2015), o servidor Proxy surgiu da necessidade de ligar a rede local à grande rede de computadores, a Internet, através de um computador que provesse o compartilhamento de Internet com os demais computadores. Pode-se fazer a seguinte analogia: rede local é uma rede interna e a internet é uma rede externa, sendo assim, o Proxy é o dispositivo que permite às máquinas da rede interna se conectarem ao mundo externo. Como na maioria dos

casos as máquinas da rede local não têm um endereço válido para a Internet, elas fazem a solicitação de um endereço externo para o servidor Proxy, que encaminha a requisição à Internet. Caso não ache o documento solicitado em seu cache de Internet, o servidor está habilitado a fazer essa consulta, pois o mesmo tem um endereço válido na Internet. Sendo assim, pode-se dizer que é normal ter um servidor Proxy diretamente ligado à Internet e com um endereço válido.

Um Proxy pode, ainda, ser definido como um software que atua como gateway de aplicação entre o cliente e o serviço que é acessado. O servidor Proxy intercepta as requisições do cliente enviadas ao servidor, as interpreta e então repassa as requisições ao servidor de destino, responsável pelo serviço a ser acessado, realizando o mesmo procedimento com a resposta.

Aprofundando-se mais no servidor Proxy, Resende e Stella (2015) ressaltam que o servidor Proxy é capaz de analisar os pacotes de rede na camada de aplicação (camada sete do modelo OSI). Dessa forma é possível oferecer uma grande flexibilidade, pois isso permite que o tráfego de dados de um serviço possa ser analisado, permitindo, assim, diversos tipos de ações, como por exemplo a aplicação de filtros de bloqueio, ou então o registro dos dados trafegados (para fins estatísticos e/ou de monitoramento, por exemplo).

Um Proxy tem a função de concentrar todas as requisições das mais diversas origens, canalizando-as por uma mesma saída. É ele que, efetivamente, faz a requisição ao destino. Funciona como um intermediário entre o cliente e o servidor de destino. Esse intermediário efetua tais requisições segundo regras, ou filtros, implementados pela ferramenta de Proxy. Tais filtros têm a função de proibir ou liberar acessos a sites, endereços identificadores de máquinas e redes, strings e até de limitar velocidade de acesso. Um servidor Proxy é utilizado para gerir os direitos de acesso, largura de banda ou distribuir conteúdo cache.

Os quatro principais tipos de Proxy são *Web, Caching, Reverter e Transparente*. Alguns servidores têm proxies que incluem várias funções, tais como Web e cache ou reverter e cache.

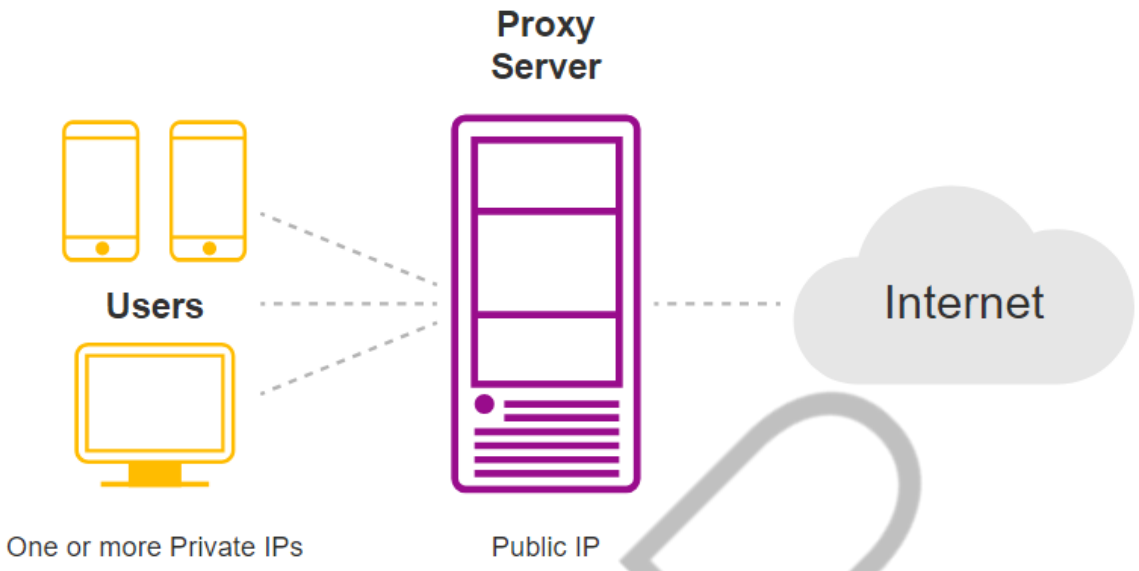

Figura 8.1 - Funcionamento de um Servidor Proxy
Fonte: Google Imagens (2020)

8.1.1 Web Proxy

O servidor ao qual você deseja se conectar só vê o servidor Proxy como sendo válido, e não o seu computador, logo seu endereço IP permanece desconhecido para o servidor, o que aumenta seu anonimato na internet.

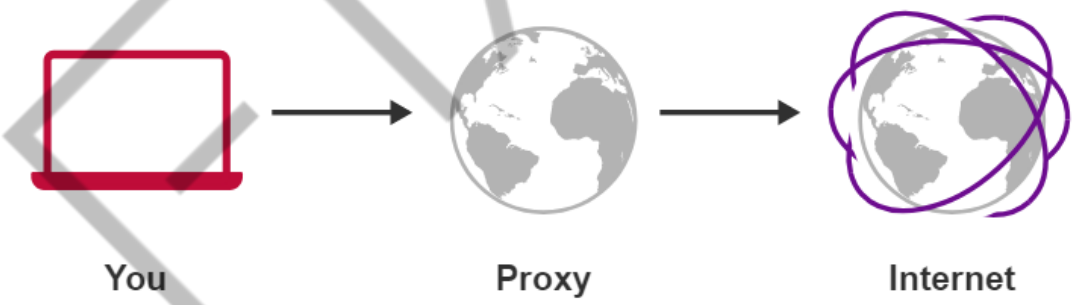

Figura 8.2 - Funcionamento de um Web Proxy para acesso à internet
Fonte: Google Imagens (2020)

8.1.2 Cache Proxy

Conforme Resende e Stella (2015), o cache é o local em que os arquivos requisitados pelo servidor Proxy são armazenados e repassados posteriormente para os clientes, que são as estações de trabalho da rede interna. Esse é um aspecto que deve ser monitorado sempre, pois pode deixar um servidor inoperante, já que são arquivos armazenados em disco e caso falte espaço em disco o servidor não vai mais

funcionar. Para que isso não aconteça é necessário determinar quando os objetos serão atualizados ou removidos do cache, sendo que alguns desses podem permanecer sem alteração alguma por tempo indeterminado e outros podem sofrer alterações frequentemente.

Visando o controle do cache, os servidores Proxy utilizam algoritmos de substituição que monitoram os objetos conforme seu cabeçalho, que contém a informação de período, tamanho e histórico de acessos. Dois deles são o *Least Recent Used*, que remove objetos existentes há muito tempo e o *Least Frequently Used*, que remove os objetos menos utilizados. A utilização do espaço em disco pelo cache do Proxy é controlada por meio desses algoritmos, juntamente com regras predeterminadas pelo administrador.

No caso de um objeto expirado, o servidor web original será consultado para revalidá-lo. Quando o objeto tem em seu cabeçalho o campo *Last-Modified*, indicando qual foi sua última alteração, o Proxy pode usá-lo para fazer a requisição *If-Modifed-Since* ao servidor web remoto, fazendo a comparação da data de alteração, identificando se o objeto foi alterado ou não e poderá atualizá-lo, caso necessário, no seu cache.

Segundo Resende e Stella (2015), existem três tipos de cache:

- Browser Cache – a maioria dos navegadores de internet possuem um cache próprio, pois é muito provável que os usuários acessem os mesmos objetos frequentemente e, nesse caso, o cache não é compartilhado;
- Proxy Cache – são as implementações mais utilizadas de Proxy, e são conhecidos também como caching web Proxy. Elas disponibilizam em cache páginas e arquivos de servidores remotos da Internet, permitindo que os clientes da rede local acessem de forma rápida esses arquivos, considerando que a velocidade do link da LAN é muito maior do que com a internet.

Quando o Proxy cache recebe uma solicitação de acesso a um recurso externo, como uma página da internet, ele procura primeiramente em seu cache local e, caso não encontre o recurso solicitado, ele imediatamente faz a requisição à Internet, armazenando em seu cache e respondendo a solicitação do cliente. Por esse motivo, pode-se afirmar que a web Proxy, além de prover segurança, provê, também, alto

desempenho para o acesso à internet e permite criar filtros, por meio de regras, dizendo o que é permitido e o que é proibido.

A aplicação Proxy age como um serviço intermediário entre as estações e os servidores remotos de internet. Eles são utilizados por corporações que desejam reduzir a banda de comunicação que utilizam com a internet.

Uma das principais finalidades de muitos proxies é o *caching*, em que ocorre o armazenamento de uma resposta obtida anteriormente para uso mais tarde, quando os clientes solicitarem o mesmo recurso. O cache retoma a resposta se acreditar que ela ainda está fresca (ou seja, o servidor de origem teria aprovado o retorno da resposta). A função de *caching* de um Proxy é opcional, ou seja, um Proxy realiza o papel de um cache além do seu papel como servidor para os clientes que estão por trás dele e como cliente para os servidores de origem.

8.1.3 Transparent Proxy

Nesse Modelo de configuração, os clientes não necessitam e nem devem configurar o uso do Proxy, pois as conexões web serão redirecionadas ao Proxy de forma transparente (automaticamente). É necessário utilizar o *iptables* (em ambiente Linux), ou outro Firewall em caso de se tratar de um ambiente Windows, por exemplo, para que basicamente as portas 80 e 443 sejam redirecionadas a porta do Proxy (geralmente, 3128). Dessa Forma, diferentemente do modelo convencional, não é necessário configurar o Proxy manualmente em todos os computadores para navegar.

De acordo com Resende e Stella (2015), essa é uma forma de obrigar os clientes a utilizarem o Proxy, ou seja, além das características do Proxy cache, ele implementa de forma transparente (por isso o nome) políticas de utilização e permite a coleta de dados estatísticos, entre outros. A transparência é implementada com a técnica de encaminhamento de portas, uma regra feita diretamente no firewall que faz o redirecionamento de todo o tráfego, por exemplo, HTTP, porta 80, para o Proxy. Sendo assim, não importam as configurações do usuário, pois sua utilização estará sempre condicionada à política de acesso predeterminada.

8.2 Autenticação do Proxy pelo LDAP

Segundo Resende e Stella (2015,) o Protocolo Leve de Acesso a Diretório ou LDAP foi desenvolvido e padronizado em julho de 1993. Com o passar dos anos, várias aplicações começaram a dar suporte de acesso a diretórios. O conceito de diretório é definido como “algo para indicar direções”. A partir desse conceito, nota-se que o serviço de diretório serve como um indicativo para localizar a informação desejada. Além de uma ferramenta eficiente na localização das informações, os serviços de diretórios devem proporcionar o gerenciamento dessas informações, tendo como princípio a centralização, pois, se existirem muitas fontes, essas informações podem estar desatualizadas. No LDAP, a recuperação de informações é iniciada pela raiz da árvore e o dispositivo de busca vai percorrendo os nós até encontrar o elemento desejado.

Conforme explica Resende e Stella (2015), a raiz e os ramos da árvore são diretórios. Cada diretório pode conter outros diretórios ou elementos que são chamados de entradas; cada entrada possui um ou mais atributos que, por sua vez, podem ter um ou mais valores associados a eles, todos de acordo com um tipo de dado predefinido.

A inserção dos dados na base LDAP é feita por meio de arquivos LDIF ou *LDAP Data Interchange Format*, de texto puro, e que permitem a importação e alteração de dados, o backup e a replicação do diretório. As principais características de serviço de diretórios são:

a) auxiliar na organização das informações, centralizando em único repositório;

b) permitir que as informações sejam gerenciadas;

c) serviços de rede podem utilizar a informação centralizada em um repositório.

8.3 Serviço de Web Proxy em ambientes Linux

Segundo Resende s Stella (2015), o serviço de Web Proxy em ambientes Linux é realizado com a ferramenta Squid. O Squid define um conjunto de regras, conhecidas também como ACL que informam ao servidor Squid os acessos ou

bloqueios que são permitidos. Tem como função a autenticação Proxy e o histórico cache dos acessos realizados.

Como Proxy, ele exerce o papel de intermediário das transações realizadas na web, pois aceita a requisição de um determinado cliente, faz o processamento e encaminha ao servidor. A requisição pode ser aceita, rejeitada ou até modificada antes que o cliente receba a resposta do servidor. Como cache, armazena o conteúdo acessado recentemente para possível reutilização, pois, caso seja acessado o mesmo conteúdo, não há necessidade de que o cache recarregue a página desde o início.

O Squid apresenta algumas características, dentre elas destacam-se:

a) permite o gerenciamento das regras de acesso e bloqueios;

b) oferece estatísticas sobre o tráfego da web;

c) permite que somente os usuários autorizados possam navegar na internet;

d) reduz a carga de processamento do servidor web por meio do cache.

O funcionamento do serviço Squid em uma rede está ilustrado na Figura “Funcionamento de um Proxy Squid”.

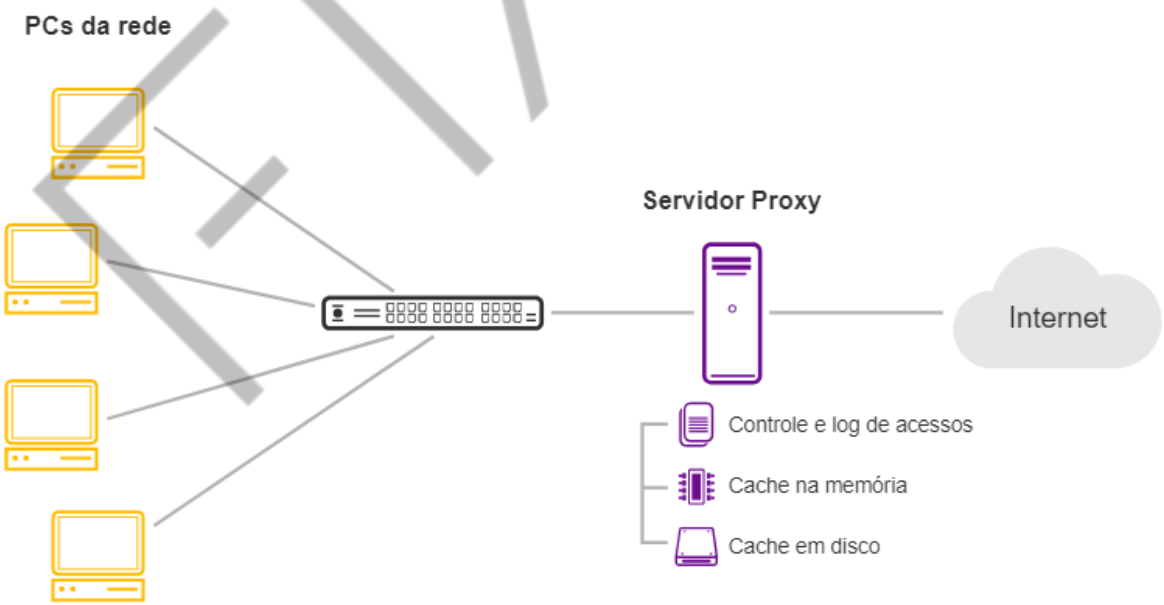

Figura 8.3 - Funcionamento de um Proxy Squid
Fonte: Google Imagens (2020)

Nesta, o serviço aceita a requisição HTTP do cliente LDAP e consulta os servidores HTTP e FTP para conceder o acesso.

O Squid é um servidor Proxy de código-fonte aberto cujo projeto iniciou-se em 1996 e utilizou-se como base o código-fonte do software *Harvest Cache Project*. O Squid é um Servidor Proxy cache de alto desempenho que suporta os protocolos HTTP, FTP, TLS, SSL. Reduz o uso da banda e melhora os tempos de resposta de páginas solicitadas que estão em cache. O Squid tem um grande controle de ACLs, sendo muito flexível. Ele aumenta a velocidade de entrega da página solicitada ao cliente. Funciona em Linux, Unix e Windows, é licenciado sob o GNU/GPL.

Resende e Stella (2015) ressaltam, ainda, que o Squid é um servidor Proxy bastante popular e oferece também a funcionalidade de cache de conteúdo, além das seguintes funcionalidades:

- Economizar banda do provedor de internet enquanto se navega na web;
- diminuir o tempo que uma página leva para carregar;
- coletar estatísticas sobre o tráfego web da rede;
- bloquear o acesso dos usuários a páginas inapropriadas, conforme a política de uso da empresa;
- garantir que apenas usuários autorizados possam navegar na internet.

Proxy quer dizer intermediário, o SQUID funciona sendo o "atravessador" entre a conexão do cliente e o servidor, neste meio do caminho ele armazena os objetos que foram solicitados e permite que as próximas requisições para os mesmos objetos possam ser respondidas por ele mesmo.

O Squid possui as seguintes características:

- Proxy e cache para HTTP, FTP e outros protocolos baseados em URL.
- Proxy para SSL.
- Cache Hierárquico.
- Suporte para Proxy Transparente.
- Políticas de controle de acesso extremamente flexíveis.
- SNMP.
- Logs Avançados.
- DNS Cache.

O Squid armazena as informações-chave de acesso dos usuários aos sites em um arquivo chamado “access.log”. Esse arquivo é baseado em linhas, ou seja, cada linha corresponde a uma requisição HTTP de um cliente.

Resende e Stella (2015) explicam que o formato padrão do arquivo de registro de acessos do Squid consiste em dez campos. A cada requisição feita por um cliente a um servidor, é armazenada no arquivo “access.log” uma linha contendo dez campos com as informações chave a respeito da conexão.

8.4 Serviço de Web Proxy em ambientes Windows

Em ambientes Windows o serviço de Proxy é realizado por um *Role Service* do Windows Server, chamado *Federation Service Proxy*. Esse serviço possui características bastante semelhantes às do Proxy Squid e pode ser habilitado em um Windows Server (2016), conforme o passo a passo:

1. Na tela inicial, digite Server Manager, e então pressione ENTER.
2. Clique em Manage, e então selecione Add Roles and Features.
3. Clique em NEXT.
4. Na próxima página selecione Role-based or Feature-based installation, e clique em NEXT.
5. Na próxima página selecione o servidor para o qual a feature será aplicada (caso possua somente um servidor em seu pool, ele já estará realçado).
6. Na próxima página, selecione Remote Access e, então, clique em NEXT.
7. Na próxima tela, selecione o check box para Federation Service Proxy e, então, clique em NEXT.
8. Na próxima tela, você deverá deixar marcada a opção para reiniciar o servidor automaticamente após a aplicação da feature. Confirme a instalação clicando em INSTALL.

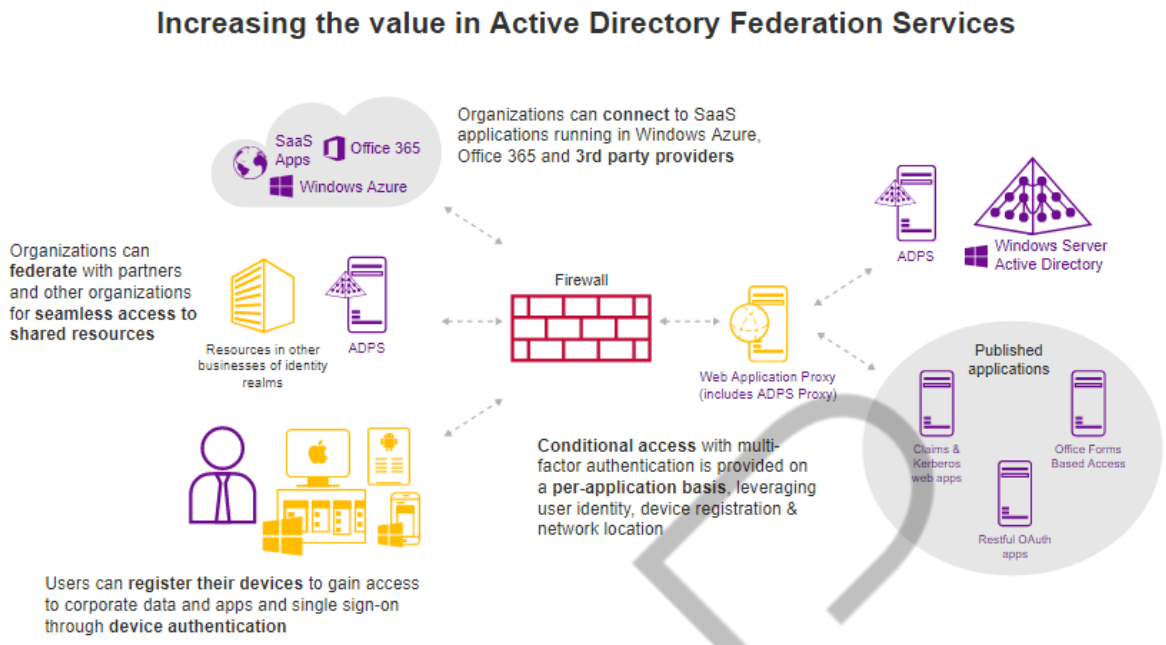

Figura 8.4 - Ambiente Windows com um Federation Proxy Service
Fonte: Google Imagens (2020)

8.5 Vantagens e Desvantagens de um Proxy

Resende e Stella (2015) destacam algumas das principais vantagens de incentivar o uso de servidores Proxy:

- Redução do tráfego de rede – são utilizadas menos requisições e respostas, sendo que o objeto do cache é recuperado, atualizado ou buscado do servidor uma única vez, o que reduz consideravelmente a utilização de banda por parte do cliente;
- Redução da carga dos servidores – são feitas menos requisições para os servidores web responderem. Por exemplo, diminui consideravelmente o congestionamento desses servidores, quando há o lançamento de um novo produto;
- Redução de latência – possibilita maior velocidade à resposta de requisições que são feitas ao objeto que está no cache do Proxy e não diretamente ao servidor remoto;
- Possibilidade de acesso – considerando que a página de internet solicitada está inacessível, se a página estiver como um objeto do cache, será possível responder à requisição, apenas não possibilitando a atualização da página solicitada.

Algumas das principais desvantagens na utilização de servidores Proxy, são:

- Poucos serviços suportados – nem todos os serviços têm suporte com os proxies atuais, sendo assim a relação entre o cliente e o servidor Proxy deve ser muito bem analisada;
- Atualização de configurações em clientes – carga muito grande de modificações e/ou atualizações em clientes, principalmente em redes locais com grande número de equipamentos. Em ambientes mistos o problema pode ser maior;
- Segurança em protocolos e aplicações – o Proxy não garante a segurança de um cliente para possíveis falhas de segurança em protocolos ou aplicações, sendo assim é necessário que o Proxy seja implementado junto a um firewall.

8.6 Lab

Neste Lab, vamos realizar a instalação e a configuração de um serviço de Proxy Transparente em ambiente Linux, utilizando para isso a aplicação Squid3 em um servidor Debian.

Para execução dos comandos abaixo utilize o usuário `root` (ou outro usuário presente no `/etc/sudoers`, sempre colocando o comando `sudo` na frente) pois serão necessários privilégios de administrador para efetuar as mudanças no sistema.

Instalando e configurando o Squid 3

Antes de tudo, vamos fazer a instalação do Squid 3. Vamos atualizar nossa lista de pacotes, utilizando um terminal, logue como usuário \"root\" ou utilize \"sudo\" antes de cada comando, sendo que, nesse caso, seu usuário precisa ter privilégio no sudoers.

Instalação dos pacotes

Vamos proceder a atualização dos repositórios do sistema e realizar a instalação dos pacotes necessários para nosso LAB:

```
apt-get update
```

Após concluída a tarefa, vamos instalar o pacote Squid 3:

```
apt-get install squid3
```

Concluída essa tarefa, vamos à configuração do Squid.

O coração do Squid fica em `/etc/squid3/squid.conf`, em que vamos fazer o seu backup e, em seguida, trabalhar suas configurações. Quando for abrir esse arquivo, use seu editor de texto preferido, eu utilizo o “Vim”, mas você pode substituir pelo que preferir.

Backup do squid.conf original:

```
cp -v /etc/squid3/squid.conf /etc/squid3/squid.conf.BACKUP
```

Apagando o original:

```
rm -v /etc/squid3/squid.conf
```

Criando nossa configuração em um novo `squid.conf` (use seu editor preferido):

```
vim /etc/squid3/squid.conf
```

Abrindo esse novo arquivo, copie e cole o conteúdo a seguir e altere para suas características. Retire meus comentários se desejar:

```
##squid.conf

http_port 3128

cache_mem 1000 MB # Se seu servidor for dedicado, coloque neste valor a metade de sua memória RAM, do contrário use apenas 25%

cache_swap_low 90
```

```
cache_swap_high 95

cache_dir ufs /var/spool/squid3 45000 16 256 # Aqui é o tamanho máximo de sua cache, no meu caso é 45GB, estude sua necessidade e capacidade da partição /var


maximum_object_size 30000 KB

maximum_object_size_in_memory 40 KB


access_log /var/log/squid3/access.log squid

cache_log /var/log/squid3/cache.log

cache_store_log /var/log/squid3/store.log

pid_filename /var/log/squid3/squid3.pid # pid - mudamos para esta pasta para facilitar na identificação de problemas

mime_table /usr/share/squid3/mime.conf


cache_mgr squid@teste.com.br

memory_pools off


diskd_program /usr/lib/squid3/diskd

unlinkd_program /usr/lib/squid3/unlinkd


refresh_pattern ^ftp:            1440    20%     10080

refresh_pattern ^gopher:        1440    0%      1440

refresh_pattern (cgi-bin|\\?)    0    0%      0

refresh_pattern .                0    20%   4320

quick_abort_max 16 KB

quick_abort_pct 95
```

```
quick_abort_min 16 KB
request_header_max_size 20 KB
reply_header_max_size 20 KB
request_body_max_size 0 KB

acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8
acl vlan24 src 192.168.30.0/24 # Representa a sua rede e respectiva máscara de sub-rede
acl lan src 172.18.28.0/24 # Assim como neste exemplo, se você tiver mais de uma rede, deve ser expressada uma por uma

acl SSL_ports port 443 563
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 563 1863 # https
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
```

```
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports

http_access allow vlan24
http_access allow vlan26
http_access allow vlan28

cache_mgr webmaster
mail_program mail
cache_effective_user proxy
cache_effective_group proxy
httpd_suppress_version_string off
visible_hostname zenhulk

error_directory /usr/share/squid3/errors/Portuguese/
```

Criar um usuário `proxy` ou outro nome se preferir, alterando no `squid.conf`:

```
useradd proxy
```

Dê permissão à pasta de log do Squid para o usuário que você criou (no Debian e em outras distros fica em /var/log/squid3/):

```
chown proxy.proxy /var/log/squid3/
```

Se você estiver com dificuldades para encontrar essa pasta, basta criá-la com essas características e dar as permissões ao usuário criado e mudar o padrão do script `squid.conf` aqui postado.

```
mkdir -p /var/log/squid3
```

Não se esqueça da permissão após criar a pasta:

```
chown proxy.proxy /var/log/squid3/
```

Se você for implementar um Proxy Transparente, deverá aplicar algumas configurações em seu script de firewall (iptables).

No arquivo `/etc/squid3/squid.conf`, na primeira linha:

```
http_port 3128
```

Mudar para:

```
http_port 3128 transparent
```

Salve e execute o comando (esse comando é muito útil para que você reconfigure o Squid sem parar o serviço):

```
squid3 -k reconfigure
```

Agora vamos criar o arquivo `/etc/init.d/firewall` com seu editor de texto preferido:

```
vim /etc/init.d/firewall
```

Depois, copie e cole o conteúdo a seguir em seu novo script de firewall:

```
#!bin/bash
echo 1 > /proc/sys/net/ipv4/ip_forward
#
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# Limite contra ping da morte e DoS
iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -m limit --limit 1/s -j DROP

# Liberando portas SSH a partir de qualquer interface
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
     # Liberando portas squid, http e NTP. Estes serviços o firewall só irá
     # responder se vierem da interface da rede interna. Daí você aplica de
     # acordo com o cenário de sua máquina
     iptables -A INPUT -p tcp --dport 3128 -i eth0 -j ACCEPT
     iptables -A INPUT -p tcp --dport 80 -i eth0 -j ACCEPT
     iptables -A INPUT -p tcp --dport 21 -i eth0 -j ACCEPT
     iptables -A INPUT -p tcp --dport 123 -i eth0 -j ACCEPT
     iptables -A INPUT -p udp --dport 123 -i eth0 -j ACCEPT

     # Nat Global (aqui você faz o mascaramento de forma geral para qualquer
     # interface ou rede do servidor. Prefiro especificar a sub-rede, deixarei
     # comentada a linha global
     #iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

     # NAT Rede 1 e 2 (substitua as respectivas sub-redes pelas que você usa em
     # sua rede e note que a \"eth1\" é minha interface diretamente conectada à internet,
     # substitua-a pela sua interface adequada
     iptables -t nat -A POSTROUTING -s 192.168.30.0/255.255.255.0 -o eth1 -j MASQUERADE
     iptables -t nat -A POSTROUTING -s 172.18.28.0/255.255.255.0 -o eth1 -j MASQUERADE
```

```
     # Jogando tráfego da porta 80 para o SQUID3 TRANSPARENT
     iptables -t nat -A PREROUTING -s 192.168.30.0/255.255.255.0 -p tcp --dport 80 -j REDIRECT --to-port 3128
     iptables -t nat -A PREROUTING -s 172.18.28.0/255.255.255.0 -p tcp --dport 80 -j REDIRECT --to-port 3128


     # Exemplo de redirecionamento de porta, neste caso redirecionamento para
     # Terminal Server e VNC (substitua as respectivas sub-redes pelas que você usa
     # em sua rede e as portas respectivas do serviço que quer redirecionar.
     # Note que a \"eth1\" é minha interface diretamente conectada à internet,
     # substitua-a pela sua interface adequada
     iptables -A FORWARD -p tcp --dport 3389 -d 182.164.2.1 -j ACCEPT
     iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 3389 -j DNAT --to 192.168.30.32
     iptables -A FORWARD -p tcp --dport 3389 -d 182.164.2.1 -j ACCEPT
     iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 5900 -j DNAT --to 172.18.28.55
```

Você pode aproveitar essas dicas para deixar seu firewall um pouco mais robusto, porém se quiser apenas que resolva o proxy transparente, basta as linhas a seguir, salvo que somente elas, sem qualquer restrição, tornam você vulnerável na internet, então, você deve aplicar suas regras da forma que lhe for conveniente, de acordo com sua responsabilidade.

Script de firewall que implementa somente o Proxy Transparente:

```
#!bin/bash

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

iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

iptables -t nat -A PREROUTING -s 192.168.30.0/255.255.255.0 -p
tcp --dport 80 -j REDIRECT --to-port 3128
```

Devemos colocar as regras que criamos no script de firewall para serem executadas sempre que o sistema inicializar. No Debian devemos adicionar as seguintes linhas ao arquivo `/etc/init.d/bootmisc.sh`:

```
vim /etc/init.d/bootmisc.sh


if [ -x /etc/init.d/firewall ]; then
    . /etc/init.d/firewall
fi
```

Então, salvamos e damos permissão para que o script seja executado pelo sistema:

```
chmod +x /etc/init.d/firewall
```

Se preferir use o arquivo `/etc/rc.local`, adicionando a seguinte linha:

```
sh -e /etc/init.d/firewall
```

Salve o arquivo e dê as permissões:

```
chmod +x /etc/init.d/firewall
```

Assim temos um Proxy Transparente com Squid3 funcionando.

REFERÊNCIAS

RESENDE, Hendrikus Francisc; STELLA, Wagner Corrêa de Oliveira. **Proxy Transparente Aplicado Em Um Servidor Institucional**. 2015. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/6465/1/PG_COADS_2015_1_03.pdf>. Acesso em: 16 jun. 2020.