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.

1. Primeiro, vamos instalar os pacotes:

```
# apt-get install squid3 sarg
```

2. Note que o arquivo de configuração do Squid é imenso, com 7655 linhas. Ele é tão grande porque inclui comentários extremamente detalhados para cada opção de configuração — excluindo-se linhas comentadas e em branco, restam apenas 24 linhas efetivas de configuração. Vamos fazer um backup do arquivo original e trabalhar apenas com o conteúdo relevante:

```
# wc -l /etc/squid3/squid.conf
7655 /etc/squid3/squid.conf

# grep -v '^#' /etc/squid3/squid.conf.orig | sed '/^$/d' | wc -l
24

# cp /etc/squid3/squid.conf /etc/squid3/squid.conf.orig
```

3. A seguir, vamos editar o arquivo de acordo com as especificações da atividade. Preste especial atenção aos blocos `http_access`, que são lidos sequencialmente de cima para baixo:

```
acl intnet1 src 192.168.0.0/24  # rede Client_Linux
acl intnet2 src 172.16.0.0/24   # rede Win7-padrao

acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # 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

# allow local networks
http_access allow intnet1
http_access allow intnet2

# default http_access block
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
http_access allow localhost
http_access deny all

# cache options
cache_effective_user proxy
cache_dir ufs /var/spool/squid3 100 16 256
cache_log /var/log/squid3/cache.log
cache_access_log /var/log/squid3/access.log
cache_store_log none

# additional configuration
http_port 3128
coredump_dir /var/spool/squid3
refresh_pattern ^ftp:   1440  20% 10080
refresh_pattern ^gopher:  1440  0%  1440
refresh_pattern -i (/cgi-bin/|\?) 0 0%  0
refresh_pattern .   0 20% 4320
shutdown_lifetime 1 second
```

4. Pare o serviço do Squid e invoque-o com a opção `-z`, que irá criar as pastas do diretório de *cache*. Após o final das mensagens de log, digite `CTRL + C` para cancelar o comando.

```
# systemctl stop squid3.service
# squid3 -z
(...)
^C
```

5. Finalmente, inicie o processo do Squid.

```
# systemctl start squid3.service
```

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.

1. Para configurar o *proxy* no Windows, acesse: Iniciar → Opções da Internet → Aba Conexões → Configurações da LAN. Desmarque a caixa "Detectar automaticamente as configurações", e marque as caixas "Usar um servidor *proxy* para a rede local" e "Não usar servidor *proxy* para endereços locais". Finalmente, insira o IP da máquina *Server_Linux* (172.16.0.10) e porta do Squid (3128) nos campos apropriados, como se segue:

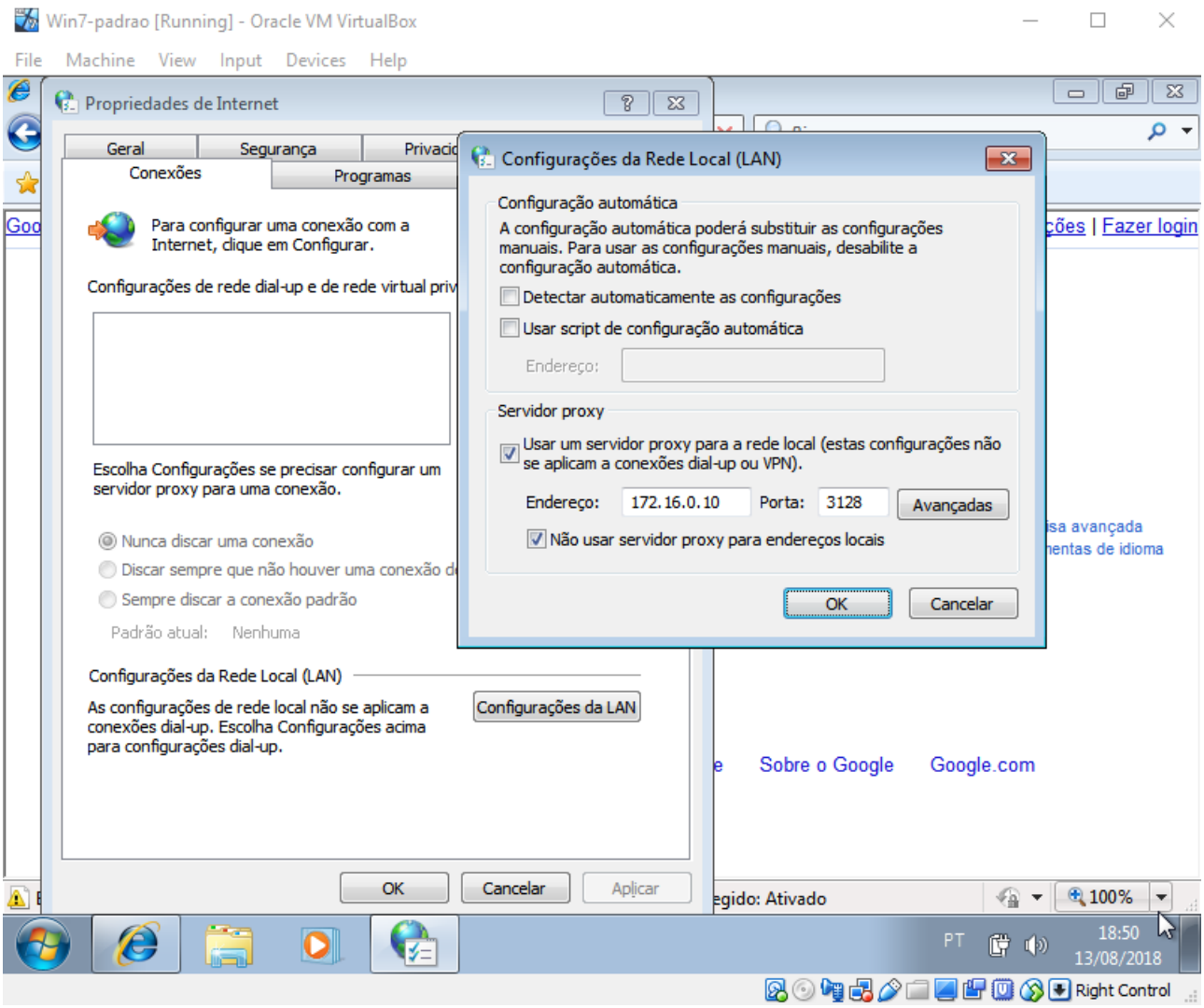

Figura 15: Configuração de proxy direto

2. Feito isso, basta acessar http://www.openbsd.org e verificar os eventos registrados no arquivo `/var/log/squid3/access.log`.

```
# tail -f -n0 /var/log/squid3/access.log
1534200817.622   2157 172.16.0.51 TCP_MISS/200 5495 GET http://www.openbsd.org/ -
HIER_DIRECT/129.128.5.194 text/html
1534200818.683   1260 172.16.0.51 TCP_MISS/200 20896 GET
http://www.openbsd.org/images/puffy63.gif - HIER_DIRECT/129.128.5.194 image/gif
1534200818.977   1323 172.16.0.51 TCP_MISS/200 50729 GET
http://www.openbsd.org/images/rack2009-s.png - HIER_DIRECT/129.128.5.194 image/png
1534200819.749    572 172.16.0.51 TCP_MISS/200 5003 GET
http://www.openbsd.org/favicon.ico - HIER_DIRECT/129.128.5.194 image/x-icon
```

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.

1. Primeiro, vamos implementar o controle por endereço físico. Edite o arquivo `/etc/squid3/acl/mac.conf` e inclua:

```
08:00:27:00:ca:5f
```

No topo do bloco `acl` do arquivo de configuração `/etc/squid3/squid.conf`, inclua a linha:

```
acl block_mac  arp           "/etc/squid3/acl/mac.conf"
```

No topo do bloco `http_access`, faça o bloqueio.

```
http_access deny block_mac
```

Recarregue a configuração do Squid:

```
# systemctl reload squid3.service
```

Na máquina *Win7-padrao*, teste a configuração acessando algum link no website http://www.openbsd.org . Como a configuração deste bloqueio reagiria em casos de *arp spoofing*?

2. Agora vamos implementar o bloqueio por IP. Edite `/etc/squid3/acl/ip.conf`:

```
172.16.0.51
```

Da mesma forma que antes, inclua no topo do bloco `acl`:

```
acl block_ip   src           "/etc/squid3/acl/ip.conf"
```

E agora, no topo do bloco `http_access`:

```
http_access deny block_ip
```

Recarregue o Squid e teste na máquina *Win7-padrao*. Como esta configuração reagiria no caso de troca do *lease* pelo servidor DHCP?

3. Vamos partir para o controle por horário. Edite `/etc/squid3/acl/time.conf`:

```
MTWHF 00:00-06:00
MTWHF 19:00-23:59
```

No topo do bloco `acl`:

```
acl block_time time          "/etc/squid3/acl/time.conf"
```

E no topo do bloco `http_access`:

```
http_access deny block_time
```

Ajuste o relógio do sistema para um horário bloqueado, como 23h por exemplo:

```
# date -s 23:00:00
# hwclock --systohc
```

Recarregue o Squid e teste o acesso na máquina *Win7-padrao*.

4. Vamos para o bloqueio de extensões. No arquivo `/etc/squid3/acl/regex_ext.conf`:

```
\.avi$
\.mp3$
\.pdf$
```

No topo do bloco `acl`:

```
acl block_ext  urlpath_regex "/etc/squid3/acl/regex_ext.conf"
```

E no topo do bloco `http_access`:

```
http_access deny block_ext
```

Recarregue o Squid e teste o acesso na máquina *Win7-padrao*. Para encontrar um arquivo que se ajuste ao bloqueio elaborado, basta pesquisar no Google algo como `site:ftp.openbsd.org filetype:pdf`, e clicar em um dos resultados.

5. Agora o bloqueio por palavras ocorrendo em URLs. Edite `/etc/squid3/acl/regex_word.conf`:

```
crypto
playboy
sexo
torrent
virus
```

No topo do bloco `acl`:

```
acl block_word urlpath_regex "/etc/squid3/acl/regex_word.conf"
```

Agora, no topo do bloco `http_access`:

```
http_access deny block_word
```

Recarregue o Squid e teste o acesso na máquina *Win7-padrao*. A URL http://www.openbsd.org/crypto.html é uma boa candidata para testar a efetividade do bloqueio.

6. Finalmente, vamos para o bloqueio por domínio. Edite `/etc/squid3/acl/domain.conf`:

```
.facebook.com
.instagram.com
.twitter.com
.whatsapp.com
```

No topo do bloco `acl`, inclua:

```
acl block_domain  dstdomain     "/etc/squid3/acl/domain.conf"
```

E no topo do bloco `http_access`:

```
http_access deny block_domain
```

Recarregue o Squid e teste o acesso na máquina *Win7-padrao*. A URL http://web.whatsapp.com é um bom exemplo do alvo do bloqueio.

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.

1. Assim como o Squid, o arquivo de configuração do SARG é imenso — 687 linhas. Dessas, apenas 43 são configurações efetivas.

```
# wc -l /etc/sarg/sarg.conf
687 /etc/sarg/sarg.conf

# grep -v '^#' /etc/sarg/sarg.conf | sed '/^$/d' | wc -l
43

# cp /etc/sarg/sarg.conf /etc/sarg/sarg.conf.orig
```

2. Vamos fazer o backup do arquivo original e trabalhar com algo mais gerenciável:

```
# mytemp=$(mktemp) && grep -v '^#' /etc/sarg/sarg.conf | sed '/^$/d' > $mytemp &&
mv $mytemp /etc/sarg/sarg.conf
```

3. Das linhas originais, precisamos alterar o valor de apenas seis, que se seguem:

```
access_log /var/log/squid3/access.log
output_dir /var/www/meusite/squid-reports
resolve_ip no
date_format e
use_comma no
charset UTF-8
```

4. Agora rode o comando `sarg`. Observe que o diretório `/var/www/meusite/squid-reports` foi criado automaticamente. Para tornar a geração de relatórios periódica, pode ser interessante agendar a execução do `sarg` no `cron` do sistema.

```
# sarg

# ls -1 /var/www/meusite/
index.html
restrito
squid-reports
```

5. Agora, basta acessar a URL `https://meusite.empresa.com.br/squid-reports/` e verificar os relatórios produzidos.

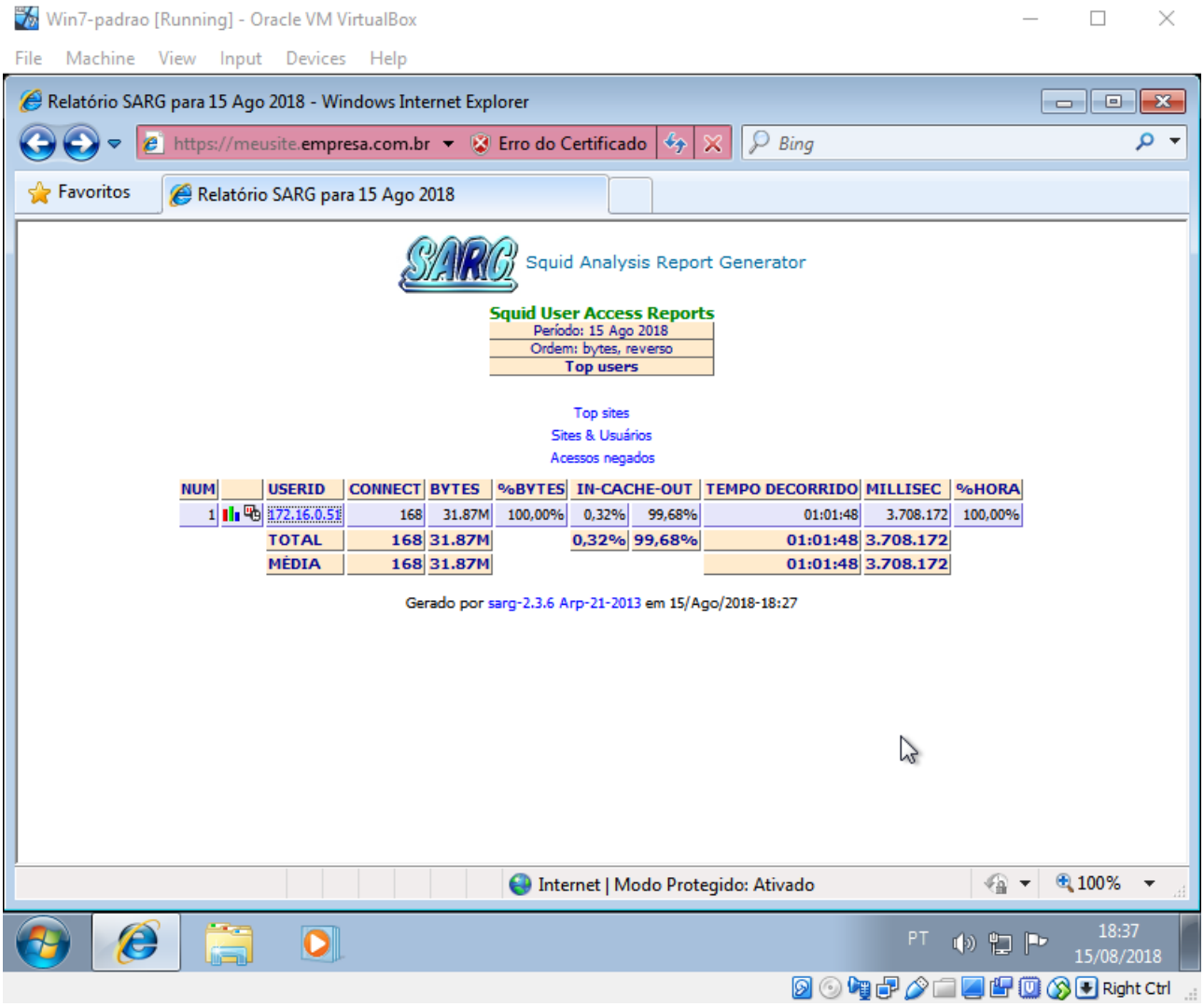

Figura 16: Visualização dos relatórios do SARG

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.

1. As configurações do firewall feitas através do comando `iptables` ficam apenas em memória, e se perdem após o `reboot` da máquina. Instale o `iptables-persistent` para corrigir isso:

   ```
   # apt-get install iptables-persistent
   ```

2. Verifique que as configurações do firewall estão vazias, exceto pelo *masquerading* que criamos na atividade inicial de configuração do laboratório:

```
# iptables -L -vn
Chain INPUT (policy ACCEPT 450 packets, 68775 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 213 packets, 53922 bytes)
 pkts bytes target     prot opt in     out     source               destination

# iptables -L -vn -t nat
Chain PREROUTING (policy ACCEPT 344 packets, 26393 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 334 packets, 25657 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 449 packets, 32617 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 77 packets, 5657 bytes)
 pkts bytes target     prot opt in     out     source               destination
  377 27216 MASQUERADE  all  --  *      eth0    0.0.0.0/0            0.0.0.0/0
```

3. Faça o `REDIRECT` de pacotes com destino à porta 80/HTTP para a porta 3128/TCP da máquina local. Depois, grave as configurações no arquivo `/etc/iptables/rules.v4`:

```
# iptables -t nat -A PREROUTING -p tcp -m tcp --dport 80 -j REDIRECT --to-port 3128

# iptables-save > /etc/iptables/rules.v4
```

4. No `/etc/squid3/squid.conf`, configure a porta 3128 em modo transparente:

```
http_port 3128 transparent
```

5. De volta à máquina *Win7-padrao*, limpe as configurações de *proxy*:

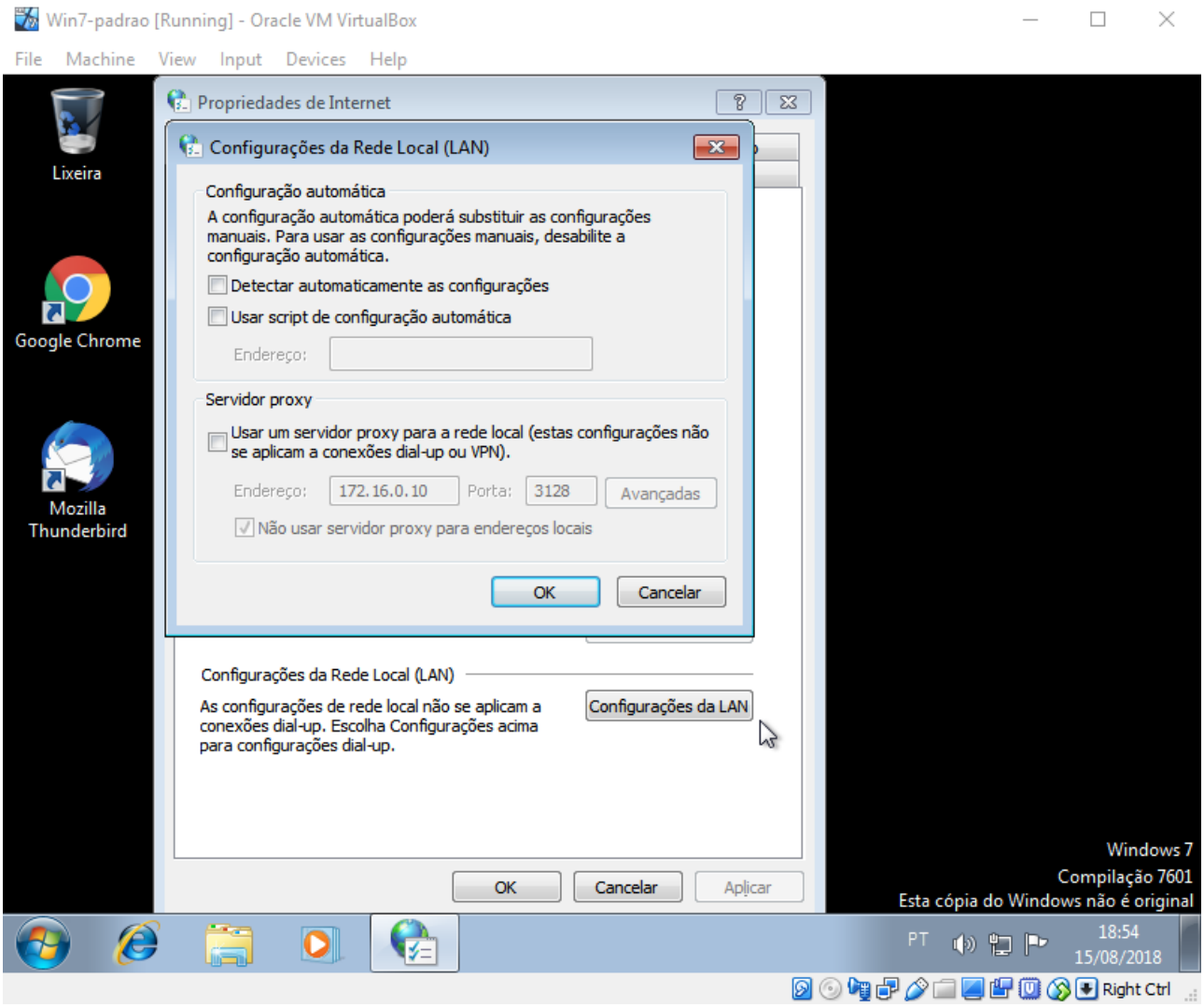

Figura 17: Configuração de proxy transparente

6. Finalmente, acesse uma URL na porta 80/HTTP e verifique se o *proxy* continua operacional.

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.