GTS-8

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

Helder Jean Brito da Silva
(helder@info.ufrn.br)

Ricardo Kléber Martins Galvão
(rk@info.ufrn.br)

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Introdução
- Necessidades
- Soluções prévias
- Solução adotada na UFRN
- Problemas
- Resultados
- Conclusão

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Introdução
  - Este trabalho visa mostrar uma solução implementada no âmbito da rede UFRN, para tentar minimizar o impacto do tráfego peer-to-peer na rede da instituição.

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Necessidades
  - Diminuir o desperdício de banda
    - O acesso do usuário comum à internet termina sendo dificultado
  - Coibir a pirataria
    - Evitar processos judiciais

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Soluções prévias
  - Bloqueio das portas utilizadas pelos aplicativos P2P
    - Problema: Os aplicativos não dependem mais de uma porta padrão.

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Soluções prévias
  - Bloqueio de todas as portas não relacionadas a serviços padrões como HTTP, FTP, etc
    - Problema: dificultaria, por exemplo, aplicações multimídia

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Soluções prévias
  - Bloqueio utilizando o módulo *match string* do iptables (para GNU/Linux)
    - Problema 1: As mensagens trocadas entre aplicativos P2P variam a cada versão
    - Problema 2: Criptografia em protocolos P2P

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Soluções prévias
  - Bloqueio utilizando o Snort com assinaturas atualizadas[1] como IDS reativo
    - Problema 1: Consumo de recursos
    - Problema 2: O módulo *flexresp* do Snort não se mostrou eficiente

[1] = Projeto Bleeding Snort
http://www.bleedingsnort.com

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Solução adotada
  - Filtragem na camada de aplicação utilizando patches específicos, com iptables
    - IPP2P (http://www.ipp2p.org)
    - L7-FILTER (http://l7-filter.sourceforge.net)
  - Controle de banda utilizando HTB

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Abordagem da solução
  - Identificar e priorizar o tráfego considerado “válido”: HTTP, SMTP, FTP, etc.
  - Dar baixa prioridade ao resto, considerando este como sendo P2P
  - Bloquear portas default de aplicativos P2P

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Identificando o tráfego que receberá alta prioridade
  - O tráfego de/para a DMZ recebe prioridade alta
  - O tráfego de/para servidores da intranet recebem prioridade alta

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Identificando o tráfego que receberá alta prioridade
  - Tráfego ICMP recebe prioridade alta
  - O tráfego HTTP (porta 80) recebe prioridade alta, mesmo que não passe por proxy, devido à custosa tarefa de analisá-lo na camada de aplicação

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Identificando o tráfego que receberá alta prioridade
  - Utilizando o patch L7-FILTER, identificar protocolos funcionando em portas de serviços comuns
    - O L7-FILTER tem assinaturas bastante precisas para tráfegos que não são P2P ou baseado em SSL

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Identificando o tráfego que receberá alta prioridade
  - Utilizando o patch IPP2P, identificar eventual tráfego P2P em portas cujo tráfego deveria ser criptografado (ex: HTTPS), ou que não possa ser detectado pelo L7-FILTER

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Exemplo de script iptables

```
# Limpando as regras e considerando marcações anteriores
iptables -t mangle -Z
iptables -t mangle -A PREROUTING -j CONNMARK
 --restore-mark
iptables -t mangle -A PREROUTING -m mark ! --mark 0
 -j ACCEPT
# ICMP = prioridade alta
iptables -t mangle -A PREROUTING -p icmp -j MARK
 --set-mark 1
```

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Exemplo de script iptables (cont.)

```
# Trafego de/para a DMZ (eth1)
iptables -t mangle -A PREROUTING -i eth1 -j MARK
 --set-mark 1
iptables -t mangle -A PREROUTING -i ! eth1
 -d ips.da.dmz -j MARK --set-mark 1
```

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Exemplo de script iptables (cont.)

```
# Trafego de/para servidores na intranet (eth0)
for servidor in $(cat servidores-intranet.txt)
do
    iptables -t mangle -A PREROUTING -s $servidor -j
  MARK --set-mark 1
    iptables -t mangle -A PREROUTING -d $servidor -j
  MARK --set-mark 1
done
```

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Exemplo de script iptables (cont.)

```
# Trafego HTTP tem prioridade alta  (este tráfego pode
   ir pelo proxy)
# eth2 é a interface externa
iptables -t mangle -A PREROUTING -i ! eth2 -p tcp
 --dport 80 -j MARK --set-mark 1
```

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Exemplo de script iptables (cont.)

```
# Marcando prioridade de diversos protocolos
# Se for o protoc. correto da porta => prioridade alta
 iptables -t mangle -A PREROUTING -i ! eth2 -p tcp
 --dport 20:21 -m layer7 --l7proto ftp  -j MARK
--set-mark 1
```

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Exemplo de script iptables (cont.)

```
# Marcando prioridade de protocolos dificeis de serem
# detectados (como streams, e baseados em SSL)
# Se for tráfego P2P naquela porta => prioridade baixa
iptables -t mangle -A PREROUTING -i ! eth2 -p tcp -m
   multiport -dports 443,465,1755 -m ipp2p --ipp2p -j
   MARK --set-mark 2
```

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Exemplo de script iptables (cont.)

```
# Marcando todo o resto como baixa prioridade
iptables -t mangle -A PREROUTING -i ! eth2 -p tcp -m
   mark ! --mark 1 -j MARK --set-mark 2
iptables -t mangle -A PREROUTING -i ! eth2 -p udp -m
   mark ! --mark 1 -j MARK --set-mark 2
```

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Exemplo de script iptables (cont.)

```
# Salvando as marcacoes
iptables -t mangle -A PREROUTING -m mark --mark 1 -j
   CONNMARK --save-mark
iptables -t mangle -A PREROUTING -m mark --mark 2 -j
   CONNMARK –save-mark
```

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Exemplo de script iptables (cont.)

```
# Bloqueando portas default, para complementar
# Portas de: eMule, Bittorrent, Gnutella, Kazaa
iptables -I FORWARD -o eth2 -p tcp -m multiport --
   dports 4242,4661,4662,6881:6889,6969,6346:6348,1214
   -j DROP
```

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Exemplo de controle de banda via HTB

```
# Indicando os dispositivos que serão controlados pelo
  htb
tc qdisc add dev eth0 root handle 1:0 htb default 10
tc qdisc add dev eth1 root handle 2:0 htb default 10
tc qdisc add dev eth2 root handle 3:0 htb default 10
```

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Exemplo de controle de banda via HTB

```
# Indicando a velocidade que cada dispositivo suporta
tc class add dev eth0 parent 1:0 classid 1:1 htb rate
   1024000kbit ceil 1024000kbit
tc class add dev eth1 parent 2:0 classid 2:1 htb rate
   1024000kbit ceil 1024000kbit
tc class add dev eth2 parent 3:0 classid 3:1 htb rate
   1024000kbit ceil 1024000kbit
```

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Exemplo de controle de banda via HTB

```
# Indicando a velocidade que cada nivel de prioridade
# oferecerá
# (O mesmo se repete para as outras interfaces)
tc class add dev eth0 parent 1:1 classid 1:10 htb rate
   1023900kbit ceil 1024000kbit prio 1
tc class add dev eth0 parent 1:1 classid 1:11 htb rate
   100kbit ceil 300kbit prio 3
```

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Um complemento útil à solução
  - O protocolo Bittorrent faz uso do protocolo HTTP em sua primeira conexão
  - Pode-se utilizar um proxy HTTP (ex: Squid) para interceptar e bloquear esta primeira requisição

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Um complemento útil à solução (cont.)
  - As portas 80 e 6969 (ambas TCP) são as mais utilizadas pelo Bittorrent em sua primeira conexão
  - Redirecionando o tráfego de ambas para o Squid, podemos bloquear o tráfego torrent nelas usando uma acl como esta:
    - ^http://[0-9a-zA-Z-.]*/announce(\.php)?\?info_hash=

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Problemas
  - Falso-positivos
    - Vários serviços utilizam portas incomuns
  - Falso-negativos
    - Versões recentes de softwares P2P utilizam criptografia
  - Aumento do consumo de recursos do firewall

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Resultados
  - Tráfego médio na rede UFRN antes da solução (em horário de expediente): 28mbps downstream / 30mbps upstream
  - Tráfego médio na rede UFRN atualmente (em horário de expediente): 24mbps downstream / 8mbps upstream

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

- Conclusão
  - A idéia de tentar combinar soluções mostrou-se eficaz, na busca de minimizar o impacto do tráfego P2P na rede, até que surja uma solução definitiva para este problema (se é que teremos uma, tendo em vista a rápida evolução dos protocolos)

Implementação de uma solução baseada em Software Livre para o controle de tráfego P2P

Perguntas?

Sugestões?