LABORATÓRIO X

ROTAS E MRTG

Documento versão 0.1

Aluno: Paulo Henrique Moreira Gurgel #5634135

Orientado pela Professora
Kalinka Regina Lucas Jaquie Castelo Branco

Outubro / 2010

Laboratório X – Monitoramento MRTG

Objetivos do laboratório

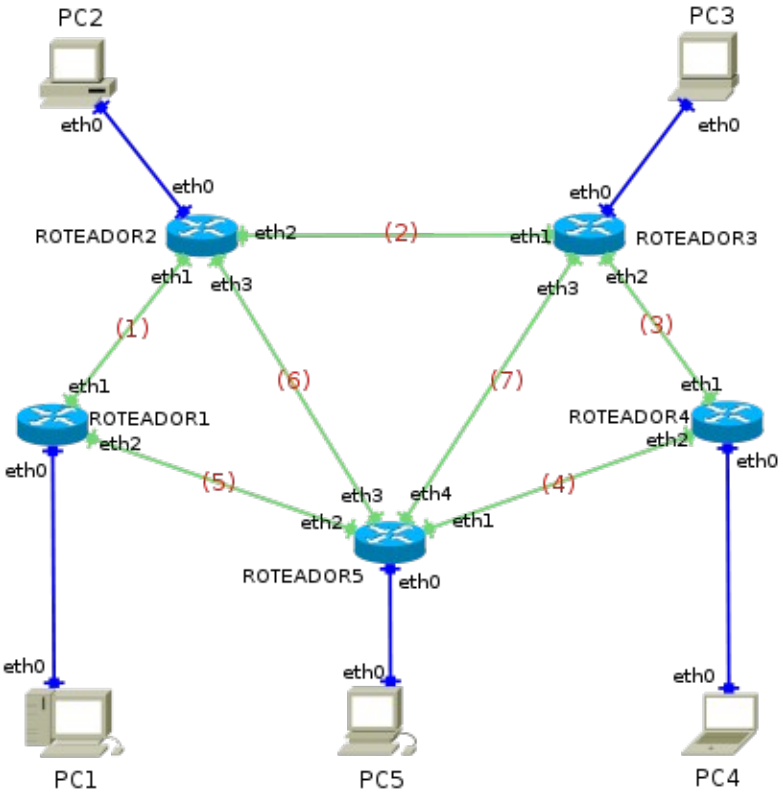
- Conhecer o trabalho de monitoramento
- Verificar o trafego que passa por uma rede
- Conhecer o Cron

Cenário sendo reproduzido

A figura abaixo representa a topologia de rede sendo estudada. Diferente das outras redes, esta precisa ser um pouco maior para demonstrar o funcionamento do protocolo RIP. Cada maquina se conecta a um único roteador. Os roteadores estão ligados entre sí de forma a permitir que exista pelo menos 2 caminhos para qualquer outro roteador.

Se um roteador cair, exceto a máquina ligada exclusivamente àquele roteador, todas as demais máquinas deverão estar em funcionamento e todas as rotas precisarão ser recalculadas rapidamente para que o funcionamento da rede não seja prejudicado.

Os números em vermelho indicam os enlaces representando uma subrede. Veja que existe um padrão estratégico na definição dos IPs.



LISTA DE INTERFACES DE REDE

```
PC1 - eth0 - 192.168.1.1/24
PC2 - eth0 - 192.168.2.1/24
PC3 - eth0 - 192.168.3.1/24
PC4 - eth0 - 192.168.4.1/24
PC5 - eth0 - 192.168.5.1/24

ROTEADOR1 - eth0 - 192.168.1.100/24
          - eth1 - 10.0.1.1/29
          - eth2 - 10.0.5.1/29

ROTEADOR2 - eth0 - 192.168.2.100/24
          - eth1 - 10.0.1.2/29
          - eth2 - 10.0.2.2/29
          - eth3 - 10.0.6.2/29

ROTEADOR3 - eth0 - 192.168.3.100/24
          - eth1 - 10.0.2.3/29
          - eth2 - 10.0.3.3/29
          - eth3 - 10.0.7.3/29

ROTEADOR4 - eth0 - 192.168.4.100/24
          - eth1 - 10.0.3.4/29
          - eth2 - 10.0.4.4/29

ROTEADOR5 - eth0 - 192.168.5.100/24
          - eth1 - 10.0.4.5/29
          - eth2 - 10.0.5.5/29
          - eth3 - 10.0.6.5/29
          - eth4 - 10.0.7.5/29
```

Conhecimentos de rede que você irá adquirir

Neste laboratório iremos utilizar a ferramenta MRTG para visualizar o tráfego de rede que passa nos roteadores por nós administrados. Estaremos utilizando num primeiro momento uma configuração muito simples e independente de SNMP.

Antes de continuar, é importante lembrar que você deve ter feito a instalação do software **Wireshark** que será utilizado neste lab, portanto use os comandos apt-get install wireshark (distribuições debian) ou urpmi wireshark (mandriva) para instalar este software, caso o mesmo não esteja instalado.

Devemos lembrar que, os comandos marcados com a tag [real] deverão ser executados no console real. Os demais comandos serão executados dentro das máquinas virtuais. Sempre que exigido a instrução pedirá uma máquina virtual específica.

Execução do laboratório

Importante: Antes de executar este lab, você desejará se prepara com os seguintes requisitos:
Este lab requer diversas janelas. Use um ambiente de trabalho com vários espaços, preferencialmente 4 deles. Gnome, Kde, Xfce tem quatro espaços por padrão. Use um deles ou configure seu ambiente preferido para quatro espaços.
Este lab não dá todos os comandos como os anteriores. Você já deve estar familiarizado com os comandos descritos e deverá executá-los sem apoio. Caso seja necessário, consulte os tutoriais anteriores.

1. [real] Salve o arquivo netkit_lab10.tar.gz na sua pasta de labs. (/home/seu_nome/nklabs).

2. [real] Acesse a pasta nklabs a partir do terminal

3. [real] Use o comando:
   ```
   [seu_nome@suamaquina ~]$ tar -xf netkit_lab10.tar.gz
   ```
   Será criada a pasta lab04 dentro da sua pasta nklabs.

4. [real] Use o comando a seguir:
   ```
   [seu_nome@suamaquina ~]$ lstart -d /home/seu_nome/nklabs/lab10
   ```

Desta vez serão iniciadas 10 máquinas virtuais, nomeadas roteador 1 a 5 e pc1 a 5. As máquinas respondem por este nome.

5. [real] Organize suas janelas de modo a localizar qualquer uma delas rapidamente. A sugestão é enviar as 5 janelas dos Pcs para o espaço de trabalho 2 e 5 janelas de roteadores para o espaço 3.

As interfaces de redes estão todas configuradas, bem como os gateways dos hosts. As tabelas dos roteadores estão limpas exceto suas rotas padrão definidas pelas próprias interfaces de rede.

Parte 1 – SNMP nos roteadores

6. Configure os serviços SNMP nos 5 roteadores, conforme aprendido no tutorial 8. Utilize o seu primeiro nome, como acesso as 3 communitys.

7. Inicie a captura com o tcpdump no roteador2, capturando a interface do ip 192.168.2.100 e salvando em sua pasta home com o nome lab10.pcap.

8. Vá ao roteador1 e inicie o daemon zebra com o comando **/etc/init.d/zebra start**.

9. Repita o procedimento com os demais roteadores.

10. Tente executar novamente o ping até o PC5 (192.168.5.1) a partir do PC1.

Vamos configurar os roteadores para exportar informações de SNMP de suas MIBs.

11. Use o comando **telnet localhost zebra** no roteador1.

12. A senha solicitada é **zebra**.

13. Pressione a tecla “?” para ver os comandos disponíveis. A tecla TAB também funciona aqui para completar comandos.

14. Dentro do zebrarot1, use o comando show ip route. Ele irá mostrar qual o caminho conhecido para atingir cada destino.

15. Use o comando **enable**. Isso irá transferir para o modo de usuário privilegiado.

16. Neste nível a senha é **zebraadmin**.

17. Use a tecla “?” e veja que há mais opções disponíveis agora.

18. Neste nível nós termos mais privilégios. Vamos trocar a senha de acesso de usuário. Use o comando **configure terminal**.

19. Use o comando **smux peer .1.3.6.1.4.1.3317.1.2.1 seu_nome** para habilitar as MIBs do roteador quagga.

20. Use o comando **exit** (para sair do modo de configuração).

21. Use o comando **disable** para sair do modo de usuário privilegiado.

22. Use o comando **exit** novamente para sair da configuração do roteador. Se você tentar entrar novamente, verá que a senha de acesso agora é teste. A senha de configuração permanece a mesma.

Aqui nós vimos a configuração do software básico de roteamento. Podemos acessar a

configuração do daemon ripd que cuida do protocolo rip. Os comandos são os mesmos. Vamos entrar brevemente.

23. Use o comando **telnet localhost ripd**

24. A senha é rip

25. Use **enable** para passar para o modo de configurador.

26. A senha é ripadmin

27. Use o comando **configure terminal** para ativar o modo de configuração.

28. Use o comando **router rip** e em seguida use o comando **smux peer 1.3.6.1.4.1.3317.1.2.3**

29. Use o comando **exit** três vezes para encerrar a configuração.

30. Acesse a pasta **/var/log/quagga** e veja o conteúdo dos arquivos ripd.log e zebra.log. (use **cat ripd.log** e **cat zebra.log**)

Os OIDS para consultar os dados são 1.3.6.1.2.1.4.24 e 1.3.6.1.2.1.23 respectivamente. Infelizmente por um bug presente na conjunto de versões quagga e net-snmp instalados no netkit, o comando SNMP-GET não obterá sucesso.

Parte 2 – Ativando o MRTG

31. Adicione as seguintes regras no firewall do roteador1:
```
iptables -t mangle -A POSTROUTING -d 192.168.0.100 -j ACCEPT
iptables -t mangle -A PREROUTING -s 192.168.0.100 -j ACCEPT
```

32. Crie o arquivo **/etc/mrtg/serv.mrtg** com o seguinte conteúdo (cuidado com copy/paste. é um comando por linha):

```
#!/bin/bash
# ip: 192.168.1.100
saida=`iptables -t mangle -L POSTROUTING -nvx | awk '/192.168.1.100/{print $2}'`
entrada=`iptables -t mangle -L PREROUTING -nvx | awk '/192.168.1.100/{print $2}'`
echo "$saida"
echo "$entrada"
```

33. Dê permissão de execução para este arquivo com o comando **chmod +x /etc/mrtg/serv.mrtg**

34. Edite o arquivo **/etc/mrtg.cfg** com o seguinte conteúdo:

```
##################################################################
# Multi Router Traffic Grapher -- Sample Configuration File
##################################################################
# This file is for use with mrtg-2.5.4c

# Global configuration
WorkDir: /var/www/mrtg
WriteExpires: Yes
```

```
Title[^]: Analise de tráfego no servidor

Interval: 5
Language: portuguese
RunAsDaemon: No

Options[_]: growright, bits, nobanner
WithPeak[_]: ymw

Target[serv]:`/etc/mrtg/serv.mrtg`
Title[serv]:"Trafego no roteador 1"
PageTop[serv]:"Trafego no roteador 1 - via IP"
MaxBytes[serv]: 300000
```

Atenção às aspas diferentes (crases) na linha target!

35. Execute o seguinte comando por três vezes seguidas, até que não apareça nenhuma mensagem de erro:
    mrtg /etc/mrtg.cfg

36. Edite o arquivo /etc/crontab e acrescente a seguinte linha no final:
    `*/5 * * * * root mrtg /etc/mrtg.cfg`

37. Reinicie o serviço cron com o comando **/etc/init.d/cron restart**.

O Cron é um serviço de tarefas agendadas, a linha acrescentada diz para o cron executar o comando de 5 em 5 minutos.

38. Precisaremos agora gerar tráfego para aparecer no gráfico. A partir do PC1, faça um **ping -s 4000** para o PC2. O parâmetro -s aumenta o tamanho do pacote.

39. Faça pings dos demais pcs para o pc1. Será necessário deixar rodando por 5 minutos. Mantenha os pings para simular uma carga.

40. Ao passar 5 minutos, copie o conteúdo da pasta /var/www/mrtg para a pasta /var/www/mrtg da máquina real. Se os gráficos estiverem em branco, use o comando do passo 35 para forçar a execução do mrtg. (dica: copie para uma pasta na hosthome e depois dela para a pasta indicada)

41. [real] Abra no browser http://localhost/mrtg/serv.html. Será necessário ter o apache instalado na máquina real. Instale-o com apt-get install apache2, se necessário.

42. [real] Use o comando a seguir para encerrar a execução do laboratório:
    `[seu_nome@suamaquina ~]$` **`lhalt -d /home/seu_nome/nklabs/lab10`**

43. [real] Use o comando a seguir para apagar os enormes arquivos.disk:
    `[seu_nome@suamaquina ~]$` **`lclean -d /home/seu_nome/nklabs/lab10`**

44. [real] Use o comando a seguir para apagar os enormes arquivos.disk restantes:
    `[seu_nome@suamaquina ~]$` **`rm /tmp/*.disk`**

45. [real] Estude a captura do tcpdump no wireshark. Você poderá usar a opção follow tcp stream para ver conteúdos inteiros.

Formule as teorias

A configuração do MRTG que vimos é muito simples. O MRTG é capaz de rodar utilizando as MIBs SNMP como fonte de informações, e rodar como um daemon. Há uma extensão chamada RRDTool que acrescenta mais configurações no MRTG.

1. Faça uma pesquisa sobre possíveis configurações do arquivo mrtg.cfg para mostrar:

a.) Utilização de processador

b.) Espaço livre em disco

c.) Memória RAM disponível

Aprendendo um pouco sobre linux

Nós completamos essa série de tutoriais em redes Ipv4 mostrando uma introdução ao MRTG, um monitor de redes. Todas as ferramentas que foram vistas até aqui podem ser utilizadas em redes IPv6 com algumas adaptações.

São possíveis trabalhos futuros:

- Servidor DHCP e static DHCP
- Proxy Squid
- NFS e Samba
- LDAP, PAM, NIS
- Outros monitores (cacti, nagios, …)
- Dynamips (Cisco IOS)
- IPv6
    - Dual Stack
    - Tunnel 6to4
    - Teredo
- MPLS