LABORATÓRIO I

UMA REDE DE DIFUSÃO SIMPLES USANDO HUB COMO DOMÍNIO DE COLISÃO

Documento versão 0.2

Aluno: Paulo Henrique Moreira Gurgel #5634135

Orientado pela Professora

Kalinka Regina Lucas Jaquie Castelo Branco

Março / 2010

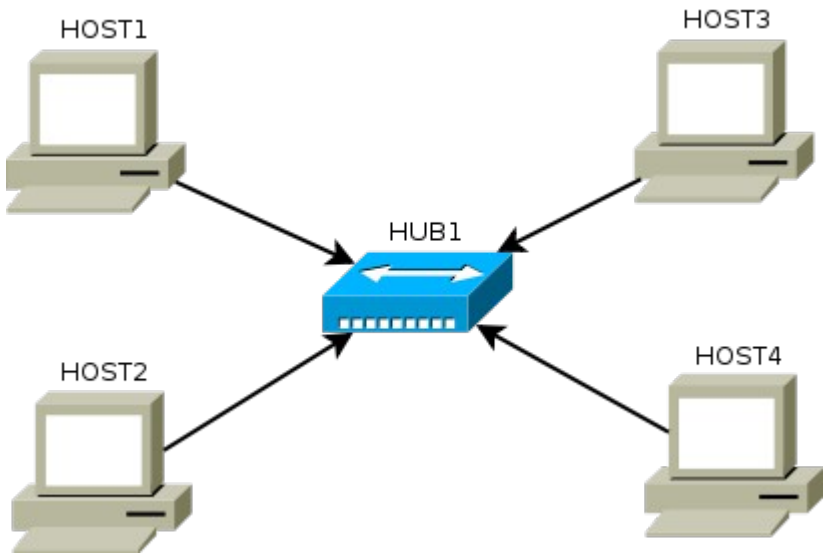
Laboratório I - Uma rede simples conectada por um hub.

Objetivos do laboratório

- Aprender como iniciar um laboratório virtual com o netkit
- Experimentar configurações de rede sobre um domínio de colisão simples
- Estudar a composição dos pacotes, desmontando os no wireshark

Cenário sendo reproduzido

A figura abaixo representa a topologia de rede sendo emulada. Estamos ligando 4 computadores nomeados host1, host2, host3 e host4 respectivamente, a um HUB nomeado HUB1. Um HUB é um dispositivo de difusão que amplifica o sinal recebido e repassa.



Seu primeiro contato com o netkit

Provavelmente você já fez uma instalação bem sucedida do netkit e deve estar ansioso para executar seu primeiro lab. Os labs serão distribuidos em pacotes compactados tar.gz e você deverá criar, preferencialmente a seguinte estrutura de pastas:

/home/seu_user/nklabs/lab01
/home/seu_user/nklabs/lab02

Onde lab01 é a pasta que conterá o lab01 e assim por diante.

Um virtual lab é composto de arquivos de configuração, principalmente lab.conf e lab.dep e das pastas que representam cada host. Nenhum deles é obrigatório no netkit. É possível não ter as pastas ou não ter os arquivos de configuração. A consequencia é que ele pode criar um lab estranho se você estiver na pasta errada e gerar vários arquivos que deverão ser apagados manualmente.

O ícone de rede ao lado, marcará um conceito sobre o funcionamento do netkit. Este conhecimento é importante para que você aprenda em breve como montar seu próprio lab.

O ícone de idéia ao lado trás a explicação sobre o conceito de redes que está sendo estudando no momento.

O ícone de exclamação ao lado alerta para algum fato importante que você deverá prestar atenção durante a execução do lab.

Durante a seção **execução do laboratório**, evite fazer experimentos para que os resultados sejam equivalentes aos da saída. Situações de erros são intencionais. A seção seguinte, **experimente**, levantará alguns questionamentos que poderão requerer experimentações e reflexões.

Conhecimentos de rede que você irá adquirir

Ao completar este lab você estará familiarizado com a configuração básica de cada host em uma rede com hub, conhecerá um pouco sobre o comando **arp** (e seu protocolo) e verá como o computador faz para localizar o endereço físico a que se destina o pacote. Descobrirá que o ICMP é um bom meio de testar a comunicação entre máquinas fisicamente ligadas e aprenderá como desmontar um pacote para visualizar seu conteúdo através do software wireshark.

Antes de continuar, é importante 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.

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

1. [real] Salve o arquivo netkit_lab01.tar.gz na sua pasta de labs. (/home/seu_nome/nklabs).

2. [real] Use o comando:
   ```
   [seu_nome@suamaquina ~]$ tar -xf netkit_lab01.tar.gz
   ```
   Ele irá criar a pasta lab01 dentro da sua pasta nklabs.

3. [real] Use o comando a seguir:
   ```
   [seu_nome@suamaquina ~]$ lstart -d /home/seu_nome/nklabs/lab01
   ```

O comando **lstart** irá iniciar o laboratório virtual, ativando o console de 4 máquinas virtuais nomeadas HOST1, HOST2, HOST3 e HOST4.

Você poderá suprimir o parâmetro -d se mudar para a pasta lab01 antes. Mas verifique com o comando pwd se a pasta corrente é lab01 antes de iniciar o lab. Caso inicie o lab em pasta incorreta sem o parametro -d, você poderá ter um lab estranho em execução e diversos arquivo para apagar.

4. [real] Use o comando vlist para listar as máquinas virtuais. Você terá como saída algo assim:

```
[seu_nome@sua_maquina lab01]$ vlist
USER              VHOST                PID       SIZE  INTERFACES
seu_nome          HOST4               1813      11604  eth0 @ HUB1
seu_nome          HOST3               2004      11604  eth0 @ HUB1
seu_nome          HOST1              32696      11604  eth0 @ HUB1
seu_nome          HOST2              32705      11604  eth0 @ HUB1

Total virtual machines:       4    (you),        4    (all users).
Total consumed memory:    46416 KB (you),    46416 KB (all users).
```

Agora você irá executar uma série de comandos, observe os resultados mostrados na tela e formule uma hipótese antes de ser apresentado à explicação exata. Alguns comandos aparentemente podem não fazer nada, travarem ou exibir mensagens de erros. Isso é normal e deve acontecer se a sequencia de passos for executada corretamente.

5. Use o comando **ifconfig** em cada uma das máquinas. Você perceberá que duas das máquinas estão com as interfaces eth0 (as placas de rede) configuradas, enquanto duas das outras máquinas não estão.
6. Execute o comando **arp** no HOST1. *(é normal não acontecer nada!)*
7. Execute, no HOST3, o comando ifconfig eth0 192.168.1.3 netmask 255.255.255.0 up.

O comando ifconfig levanta a interface de rede eth0 (primeira placa de rede) com o ip passado por parametro logo após e a máscara de subrede após o termo netmask. O termo up ativa a interface logo após a execução do comando.

8. Nos hosts 2, 3 e 4, use o comando **cd /hosthome**
9. Nos hosts 2, 3 e 4, use o comando **tcpdump -i eth0 -w lab1_hostX.pcap** (onde X é o número do host).
10. Na tela do host1, execute o comando **ping 192.168.1.2**, aguarde o resultado de alguns pings e use Ctrl + C para interromper o ping.
11. Tente executar o comando **ping 192.168.1.51** e ao receber algumas respostas cancele o comando novamente com Ctrl + C.
12. Nos hosts 2 e 3, use o comando Ctrl + C para interromper o tcpdump.
13. Use o comando **arp** em cada um dos 4 hosts e veja a saída. (se demorar para executar essa instrução a saída poderá ser diferente, deverá ter duas entradas na tabela arp da host1).
14. [real] Em sua pasta home (/home/seu_nome/) deverá existir os arquivos lab1_host2.pcap e lab1_host3.pcap. Inicie o software wireshark e abra estes arquivos. Estude seu conteúdo.

Dentro de uma máquina virtual netkit, a pasta **/hosthome** será mapeada para a pasta home do usuário que está executando a máquina virtual. Isso permite passar arquivo para dentro e fora da máquina virtual de forma simples. No caso, ao fazer cd/hosthome no passo 8, fizemos que o tcpdump colocasse o arquivo de saída numa pasta do sistema real.
Por enquanto essa é a única forma, e também é a mais simples, de enviar um arquivo para fora da máquina virtual.

Experimente

1. Levante a interface eth0 do host4, com o ip 192.168.2.51 e máscara de subrede 255.255.255.0. Levante o tcpdump nessa máquina e tente a partir de qualquer outro host, “pingar” este endereço e veja se responde.
2. Use o comando ifconfig para trocar o ip do host3 para 192.168.2.50 (basta levantar a interface novamente com o novo ip). Tente verificar com o ping a comunicação das máquinas de cada host para os demais.

Formule as teorias

Lembrando a especificação da rede, com seus atuais conhecimentos de rede, tente explicar:

1. Porque o arquivo host3.pcap parece similar ao host2.pcap? (use as informações do cenário e seu conhecimento da disciplina de redes até agora para explicar o efeito).
2. Baseado na sua explicação acima, quais são as desvantagens do uso da topologia indicada em relação à segurança da informação? E ao desempenho?
3. Porque você acredita que os ips diferentes impediram a comunicação? Dica: usando o tcpdump, você pode ver o host1 perguntando pelo ip 192.168.1.51 nos dumps dos hosts 2 e 3.

Aprendendo um pouco sobre o netkit

Você viu que os hosts 1 e 2 vieram com as interfaces de rede pré-configuradas, enquanto os hosts 3 e 4 não. Como isso é feito? Na pasta do lab, há dois arquivos chamados HOST1.startup e HOST2.startup. O HOST1.startup tem exatamente o mesmo comando ifconfig que você utilizou para levantar as outras interfaces de rede.

O HOST2.startup tem o comando **/etc/init.d/networking restart** que força o sistema a reiniciar as configurações de rede. Mas como ele sabia o ip? Navegando pelas pastas do lab, vemos que a única pasta que tem arquivos é a HOST2. Dentro da host2 temos o arquivo /**HOST2/etc/network/interfaces** que possui a configuração de rede. Na máquina virtual use o comando cat para ve-lo, na máquina real pode usar qualquer edito.

Tudo que estiver nas pastas é copiado para a máquina virtual **na primeira execução**, em referencia à pasta raíz. Após a criação do HOST.disk, o conteúdo da pasta HOST é ignorado.

Se consultar o tamanho do arquivo HOSTX.disk, verá que cada um deles o tamanho surpreendente de 10Gb, mas na verdade, apenas poucos Kbytes estão sendo realmente consumidos.

Para eliminar todo o espaço adicional consumido pela execução do lab, use o comando **lclean.** Ele irá restaurar o lab ao estado inicial, eliminando os arquivos de disk e logs. É possível configurar a interface de rede dentro da própria máquina virtual como uma distribuição linux comum, no entanto é importante enfatizar que ao excluir o arquivo.disk, todas as alterações no sistema de arquivos serão perdidas.

Por este motivo, nas práticas, você deverá aprender a usar a pasta dos hosts dos labs e os arquivos .startup para executar scripts e copiar arquivos de configuração para as máquinas virtuais.