Ficha 04 - Monitorização e diagnóstico com ICMP, UDP e TCP

Ano Letivo de 2019/2020

Cenário de testes:

Nesta ficha pretende-se abordar a utilização de aplicações populares disponíveis no Linux, para monitorização e diagnóstico das comunicações com recurso aos protocolos de transporte UDP e TCP. Algumas dessas aplicações poderão ainda não estar disponíveis no sistema operativo Linux, devendo ser instaladas, seguindo as instruções apresentadas ao longo da ficha.

O cenário de testes a utilizar consiste em duas máquinas virtuais (ou VM, *Virtual Machines*) com o Linux Ubuntu, ligadas através de uma rede interna (10.5.0.0/24), tal como ilustra a Figura 1. Para obter a segunda VM bastará efetuar um clone da existente (que o aluno já utiliza), selecionando a opção para efetuar *reset* aos endereços MAC das interfaces, tal como ilustra a Figura 2. Cada uma das máquinas mantém também o interface para uma rede externa, já existente na máquina virtual original.

Nos testes a efetuar, uma das VM poderá funcionar como cliente e a outra como servidor. Para tal, deverá ser configurada uma rede virtual para suportar as comunicações entre as duas VM, tal como ilustra a Figura 1:

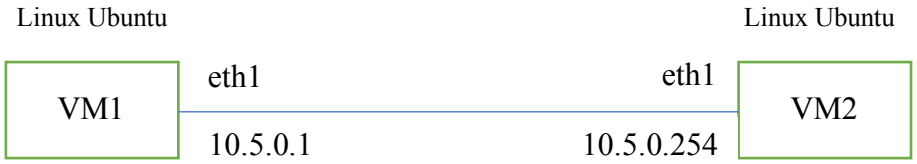

Figura 1 – Cenário experimental considerado na ficha (só é mostrada a rede interna)

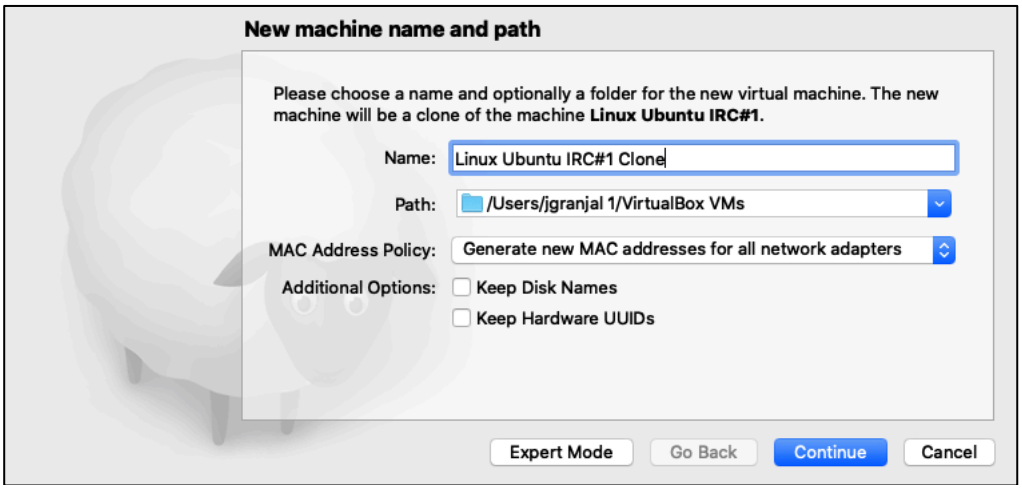

Figura 2 - Criação de VM clone (com *reset* aos endereços MAC das interfaces de rede)

Exercício 1 (configuração de cenário experimental):

Configure o cenário de rede ilustrado na Figura 1, no qual os dois sistemas Linux comunicam através de uma rede interna. Tal como ilustrado na Figura 1, a rede 10.5.0.0/24 é utilizada para endereçar essa rede interna. Para configurar a rede interna (virtual) deverá ativar, no VirtualBox, uma segunda interface de rede associada a uma "Internet Network" (ver Figura 3), que deverá ter o mesmo nome nas duas VM – esta nova interface só pode ser criada com a máquina desligada (máquinas a executar ou em estado suspenso não permitem esta operação). De seguida terá de configurar em cada uma das máquinas o endereço IP respectivo. Valide que as duas VM conseguem comunicar através desta rede virtual, recorrendo ao comando "ping".

Nota: as VM deverão continuar a dispor de conectividade com a Internet, através da primeira interface de rede (a original).

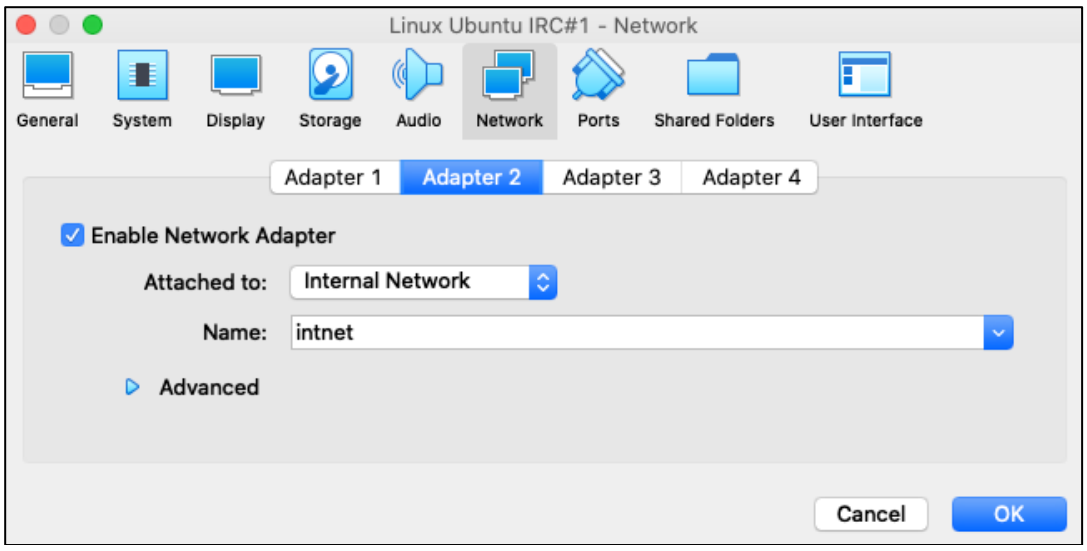

Figura 3 - Associação da segunda interface à rede interna com nome "intnet"

Exercício 2 (testes com ICMP):

Neste exercício aborda-se a utilização de duas aplicações que recorrem ao protocolo ICMP (Internet Control Message Protocol), o `ping` e o `traceroute`, bem como do `Wireshark` para capturar e analisar as comunicações produzidas por tais aplicações. O `ping` usa o ICMP para determinar se é possível comunicar com um *host* de destino e, por sua vez, o `traceroute` permite obter a rota que os pacotes IP percorrem entre o host local (de origem) e destino.

1. Recorra ao comando `ping` para validar que o host consegue contatar o servidor que responde pelo nome www.google.com. Em simultâneo, capture as comunicações geradas pela `ping` com recurso ao `Wireshark`. É provável que os pacotes ICMP sejam precedidos por pacotes utilizados pela aplicação DNS. Identifique, no conteúdo das mensagens capturadas pelo `Wireshark`, o endereço IP correspondente ao site acedido (www.google.com). Interprete os resultados obtidos (pacotes capturados pelo `Wireshark`).

2. Recorra ao comando `traceroute` para obter a rota até ao servidor que responde pelo nome www.nasa.gov. Em simultâneo, capture as comunicações geradas por esse comando. Novamente, é provável que essas comunicações sejam precedidas por pacotes utilizados pela aplicação DNS. Interprete os resultados obtidos (pacotes capturados pelo `Wireshark`).

Nota: Caso seja necessário poderá instalar no Linux as aplicações atrás referidas, por exemplo:

```
sudo apt-get install traceroute
sudo apt-get install wireshark
```

Nota 2: Para estes testes utilize apenas uma das máquinas virtuais.

Exercício 3 (análise de comunicações em interfaces de rede):

O `Ntopng` (https://www.ntop.org/) é uma aplicação que permite monitorizar, em tempo real, as comunicações de entrada e saída nas interfaces de um servidor Linux, permitindo igualmente visualizar diversas estatísticas obtidas de tais comunicações, com recurso a uma página *web*. No presente exercício iremos utilizar o `Ntopng` para efetuar alguns testes às comunicações entre os dois servidores do cenário (ver Figura 1). Assim, utilize o `Ntopng` para:

1. Obter informação sobre as comunicações ativas e registadas nas duas interfaces de rede do servidor (de ligação à Internet e à outra VM). O `Ntopng` regista, para tal informação, os endereços IP envolvidos, os portos de origem e destino e o protocolo de rede e transporte (ICMP, TDP ou UDP).

2. Obter informação relativa ao tráfego HTTP recebido pelo servidor *web* local (nota: poderá ser necessário instalar no Linux o servidor Apache, ver nota abaixo).

Nota:
Caso as aplicações atrás referidas não estejam já instaladas, use os comandos seguintes:

```
sudo apt-get install ntopng
sudo apt-get install apache2
```

Após a instalação do `Ntopng`, o serviço fica imediatamente ativo e disponível para acesso através de um interface web disponível no porto `3000` da máquina.

Exercício 4 (medições de performance com UDP e TCP):

O `iPerf` (https://iperf.fr) é uma aplicação que permite diagnosticar o desempenho das comunicações com recurso aos protocolos TCP e UDP, em diferentes condições de transmissão.

Para instalar o iperf no Linux (Ubuntu) faça:

```
sudo apt-get install iperf
```

Neste exercício iremos avaliar a performance obtida na transferência de informação entre as duas VM do cenário (ver Figura 1), de acordo com as condições descritas a seguir.

1. Recorrendo ao `iPerf` avalie a performance do UDP (taxa de transmissão e eventuais perdas de pacotes) nas comunicações entre o cliente e o servidor.

   No exercício seguinte iremos recorrer ao jPerf, uma interface gráfica para o iPerf (https://code.google.com/archive/p/xjperf/downloads), ilustrado na Figura 4 [1]. Mais informação sobre a instalação desta aplicação pode ser obtida em http://linuxthrill.blogspot.com/2016/04/how-to-install-use-iperf-jperf-tool.html.

   Nota:
   Pode ser necessário instalar também o Java: `sudo apt-get install default-jre`

2. Considere agora a utilização do TCP para suportar as comunicações entre os mesmos sistemas. Avalie o desempenho do protocolo recorrendo ao `jPerf`, variando o tamanho da janela de recepção utilizada na ligação.

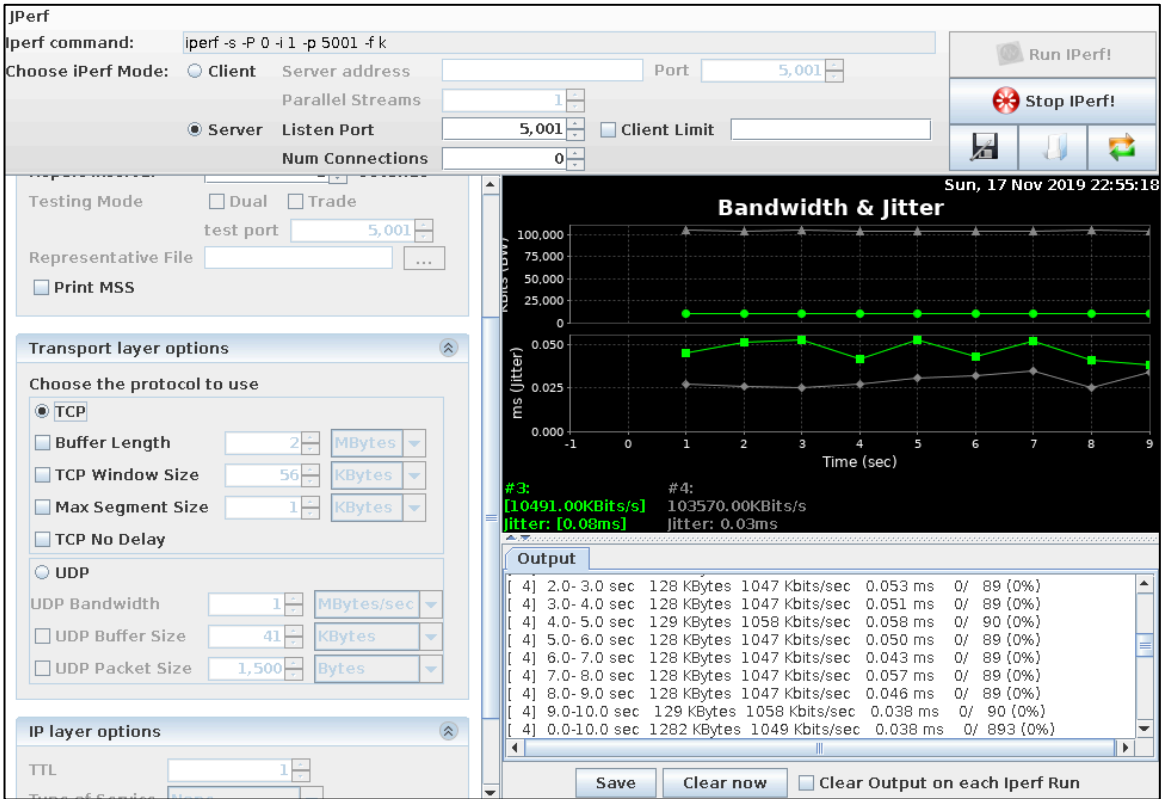

Figura 4 - jPerf, uma interface gráfica para o iPerf

[1] Mais informação sobre a instalação do jPerf pode ser obtida em: http://linuxthrill.blogspot.com/2016/04/how-to-install-use-iperf-jperf-tool.html.