1. Introdução

Este exercício destina-se a promover competências básicas na administração do sistema. Após realizar este conjunto de exercícios deverá ficar com uma visão básica daquilo que é a administração de um sistema de computação, para que serve e porque deve ser uma "tarefa sempre activa" nas atribuições da pessoa que se responsabiliza por manter um sistema deste tipo em bom funcionamento.

2. Administração de Sistema

Ser administrador de um sistema Linux, ou de qualquer outro sistema de informação, é um papel a que necessariamente está associada alguma responsabilidade, já que da actividade de administrador dependem aspectos tão importantes como a segurança do sistema e da informação que ele contém, bem como a sua correcta operacionalidade; garantir o acesso a um universo de utilizadores que pode variar ao longo do tempo, e que podem ter diferentes privilégios, garantir que o subsistema de disco opera nas melhores condições, garantir a instalação de novos periféricos ou de mais discos, garantir que há uma politica de cópias de segurança devidamente implementada, são tudo tarefas que se atribuem aquilo que vulgarmente designamos por administração de sistema.

O utilizador que garante a administração de sistema é sempre um super-utilizador, no sentido em que possui um perfil de acesso ao sistema que lhe dá poderes para fazer praticamente qualquer coisa, inclusive destruir a coerência lógica do sistema, se não souber o que está a fazer.

Assim, uma administração deficiente pode provocar sérios problemas a um sistema. Por essa razão apenas um super-utilizador a pode fazer, e é uma regra de bom senso que mesmo o administrador de sistema faça *login* como super-utilizador apenas quando é estritamente necessário, voltando a utilizar um *login* com menos privilégios em condições normais. É que não se trata apenas de falta de competências. Um engano (e eles acontecem) também pode ser fatal!

3. Comandos para explorar nesta fase

Vão ser úteis alguns comandos que lhe permitirão perceber alguns aspectos relevantes da estrutura do sistema de ficheiros, bem como ter acesso a alguma informação útil para tarefas básicas de administração.

Comece então por executar e explorar o seguinte conjunto de comandos, procurando interpretar as respostas do sistema a cada um.

Registe no seu *log book* os resultados obtidos e as suas anotações.

a) ls

b) cd (ex: ls; cd /etc; ls)
c) pwd
d) more (ex: more /etc/passwd)
e) cat (ex: cat /etc/passwd | more) (pode ter que pressionar Ctr-C para sair do comando)

4. Execução de comandos em modo root

Há ainda um comando que vamos considerar nesta fase, que é fundamental para executar tarefas de administração.

Com efeito, e em função daquilo que já foi anteriormente referido, as tarefas de administração só podem ser feitas por utilizadores especiais, os super-utilizadores, ou *root users* como também são conhecidos.

Normalmente um administrador trabalha no sistema com o seu *login* de utilizador "comum", e apenas utiliza os seus privilégios de administrador quando absolutamente necessário. Para o efeito ele utiliza o comando **sudo** (*substitute user do*) que permite a um utilizador correr programas com privilégios de outro utilizador.
O comando **sudo** pede uma *password* que corresponde à *password* do utilizador que executa o comando.

Na implementação de Ubuntu que estamos a utilizar, o utilizador por defeito tem acesso aos privilégios de administrador através do comando **sudo**. Isso pode no entanto não acontecer com todos os utilizadores. É tarefa do administrador definir esses privilégios.

5. Gestão de Utilizadores

Em ambiente de utilização profissional, a gestão de utilizadores é uma tarefa corrente na administração de sistema, uma vez que haverá um universo de utilizadores com diferentes características e necessidades que varia ao longo do tempo. Haverá de vez em quando a necessidade de criar novos utilizadores, bem como a necessidade de retirar do sistema utilizadores que já não existem.

Os sistemas Linux guardam no ficheiro **/etc/passwd** a lista de todos os utilizadores do sistema. O nome do ficheiro sugere que as *passwords* se encontram aqui, e com efeito assim era nas versões iniciais deste tipo de sistemas. Entretanto por questões de segurança associadas à evolução nos sistemas de cifra, as *passwords* mudaram de local, encontrando-se noutro ficheiro, devidamente cifradas, a que apenas o super-utilizador tem acesso.

Adicionalmente o Linux guarda a relação dos grupos de utilizadores no ficheiro **/etc/groups**.

A numeração de grupos e de utilizadores não obedece a nenhuma regra. É habitual utilizar múltiplos de cem para designar grupos (100, 500, 1400,…) e utilizar as sequencias numéricas entre grupos para designar utilizadores (601, 602, 603,…).

Explore a estrutura dos ficheiros **/etc/groups** e **/etc/passwd**, com o auxílio dos comandos já estudados, em particular os comandos **cat** e **more**, assim como a sua associação através do operador "|" que permite redireccionar a saída de um comando para a entrada do seguinte (operador *pipe*).

Acrescente um novo grupo de utilizadores ao sistema, e adicione um novo utilizador a esse grupo.

Explore os comandos e anote as suas impressões no seu *log book*:

a) addgroup
b) delgroup
c) adduser
d) deluser

6. Redes de Computadores

6.1 Introdução

Pelo menos de forma intuitiva já todos temos a noção daquilo que é uma rede de computadores. Trata-se com efeito de um conjunto de várias máquinas ligadas entre si de alguma forma, que permitem trabalhar de um modo cooperativo, partilhar recursos, comunicar, etc.

Percebe-se facilmente o impacto que trabalhar em rede poderá ter nas tarefas de administração de sistema já analisadas.

Entretanto, o facto de se trabalhar em rede obriga a precauções especiais no que diz respeito a segurança, como veremos até ao final deste semestre.

Para que numa rede os computadores interligados possam comunicar entre si, eles têm antes de mais de se "entender" perfeitamente, para o que é necessário que falem todos uma linguagem com regras comuns conhecidas por todos. A isto chama-se protocolos, e no caso dos computadores o mais vulgar é o TCP/IP (na realidade dois protocolos, o TCP – *Transmission Control Protocol* – e o IP – *Internet Protocol*).

Em matéria de comunicações, genericamente utiliza-se um modelo de comunicações designado por modelo em camadas, em que existem diversos protocolos organizados de forma hierárquica para garantir que tudo funciona, independentemente do hardware que compõe os vários computadores, do sistema operativo que corre em cada um deles etc.

O TCP/IP está algures no meio da hierarquia, sendo o TCP responsável por construir **pacotes** a partir de ficheiros ou de informação fornecida pelas aplicações, e o IP responsável por garantir o encaminhamento desses pacotes através de múltiplos nós da

rede, com base nos endereços, vulgarmente referidos por endereços "ip"[1]. Há outros protocolos de transporte igualmente utilizados e conhecidos, como é o caso do IPX para implementações de redes Novell ou do NetBEUI para implementações de redes Microsoft. Actualmente o TCP/IP é o mais implantado à escala global.

Para além da questão dos protocolos, num ambiente em que temos mais do que uma máquina, torna-se necessário diferenciar cada uma das máquinas que se encontra na rede, para o que se lhes atribui um endereço. Desta forma, com o conceito de endereço, já será possível enviar informação da máquina com endereço A para a máquina com endereço B. Neste contexto, "enviar informação" significa enviar pacotes de dados, já que a informação a transferir de uma máquina para a outra é "empacotada" em pacotes pequenos que circulam através da rede. A vantagem de ter a informação empacotada, é que se tivermos de transferir 10GB de informação entre as duas máquinas e se houver um erro na transmissão (detectado pelo protocolo de transporte) não precisamos de transferir novamente toda a informação, mas apenas o ou os pacotes onde ocorreu o erro, tornando a acção muito mais rápida.

Para além disso, o facto de utilizar pacotes, permite que o meio físico de transmissão seja partilhado por vários utilizadores, o que significa que em qualquer instante podem circular pacotes de diferentes proveniências para vários destinos, o que torna a utilização do meio muito mais eficiente.

6.2 Endereços IP

Os endereços IP associados a cada máquina em rede possuem uma estrutura muito bem definida, e se as regras a ela associadas não forem cumpridas, isso pode significar a impossibilidade total de comunicar.

Neste contexto falaremos apenas de IPv4 (versão 4), ainda a mais utilizada, sendo que a nova versão de nomenclatura para endereços a IPv6 está já em franco crescimento nos equipamentos activos da Internet.

Seja qual for o dispositivo (computador, impressora, router etc), cada endereço é composto por quatro números decimais, compreendidos entre 0 a 255 ou, em hexadecimal, 00 a FFh. Na prática estamos então a falar de endereços de 32 bits (4 x 8 bits), organizados naquilo que é vulgarmente conhecido por *dotted quad notation* (e.g. 192.168.0.1)

Há duas noções fundamentais para se poder perceber esta notação:

i) um endereço IP tem de ser único na rede
ii) parte do endereço identifica a rede onde o dispositivo se encontra, e outra parte do endereço identifica a máquina nessa rede.

Ora a questão que se coloca agora, é como é que se distinguem num endereço as duas partes referidas em ii)?

[1] O estudo detalhado deste protocolos e dos mecanismos de encaminhamento na rede, cai fora do âmbito desta UC e será estudado, como detalhe adequado, numa UC dedicada.

A resposta a esta pergunta é um outro número de 32 bits, designado por máscara de rede (*subnet mask*), que nos dá uma indicação clara daquilo que é endereço de rede e aquilo que é endereço de máquina.

A análise desta matéria daria para escrever muitas páginas sobre o tema, e está fora do âmbito deste texto. Poderá encontrar mais informação sobre o tema em http://en.wikipedia.org/wiki/IP_address, para começar a compreender o assunto.

Vamos ver de seguida alguns exemplos, cuidadosamente escolhidos para serem simples!

Exemplo 1:

IP: 192.168.42.77
Mask: 255.255.255.0

Endereço da máquina: 77
Endereço da rede: 192.168.42

Exemplo 2:

IP: 130.16.140.220
Mask: 255.255.0.0

Endereço da máquina: 140.220
Endereço da rede: 130.16

Exemplo 3:

IP: 1.10.10.10
Mask: 255.0.0.0

Endereço da máquina: 10.10.10
Endereço da rede: 1

Exemplo 4:

IP: 142.251.2.3
Mask: 255.255.240.0

Uppss!!
E agora?
Procure descobrir o problema deste 240 no terceiro número, em vez de 255 e comente. Tenha só em atenção que se fizer a representação em binário destes números, talvez seja mais fácil interpretar o que se está a passar.

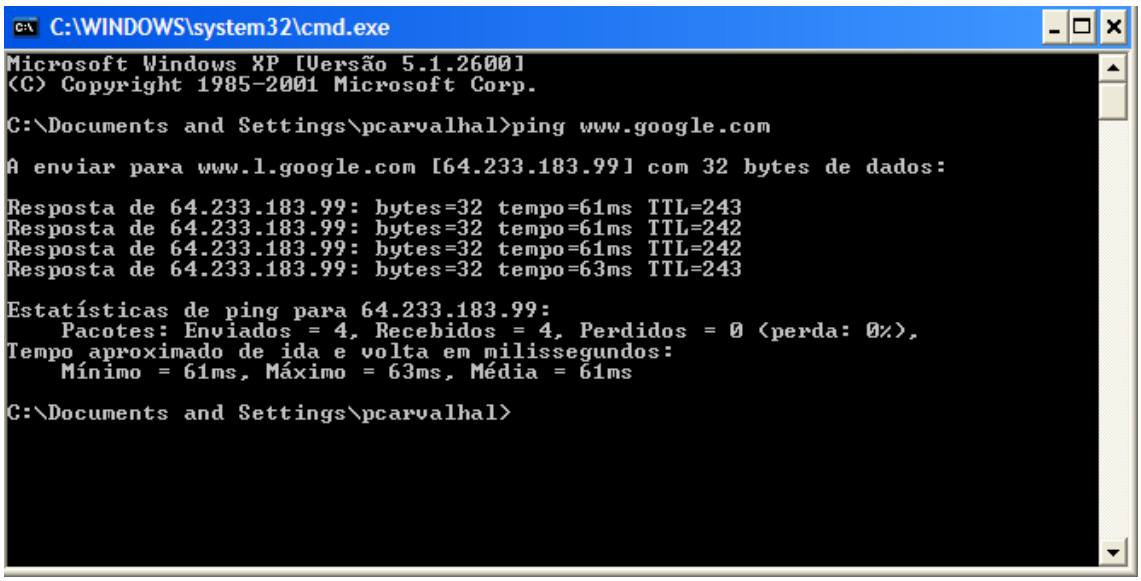
Um outro conceito fundamental associado ao protocolo TCP é o de **porta**. Como é sabido, existem muitos serviços que são assegurados pelo TCP/IP. O HTTP, o FTP, o

E-mail (na realidade são dois serviços, um para enviara e outro para receber), isto para nomear apenas os mais conhecidos. Se o mesmo servidor, como o mesmo endereço IP e com a mesma pilha de protocolos TCP/IP pode assegurar tantos serviços, como é que serão diferenciados? A resposta a esta questão é dada utilizando uma "porta", identificada por um número, a cada serviço. Essa associação é assegurada pelo TCP, que irá garantir a entrega dos pacotes destinados a uma determinada porta, à aplicação associada. Por exemplo, os pacotes destinados à porta 80 de um servidor são entregues pelo TCP ao programa servidor HTTP nele instalado.

Sendo assim, quando um servidor é criado, é necessário que o seu endereço seja conhecido ao nível da rede, assim como as "portas" que ele mantém abertas (serviços que fornece). Os serviços mais vulgares têm números de "portas" bem conhecidos (e.g, FTP porta 21; Telnet porta 23; SMTP porta 25; HTTP porta 80; POP3 porta 110; etc.).

6.3 Nomes

Experimente executar um comando **ping** no seu computador, utilizando a consola em Linux ou a janela DOS em Windows, e especificando algum servidor, como o que se mostra a seguir:



Sobre o comando **ping** falaremos noutra altura. Para já, observe que "alguém" na rede percebeu que www.google.com corresponde ao endereço IP 64.233.183.99.

O que se pode concluir é que ao endereço IP se encontra associado um nome. Sem dúvida, é mais fácil fixar o nome que o endereço!

6.4 Firewall

O conceito de *Firewall* é muito simples de entender. Prende-se fundamentalmente com a capacidade de filtrar os pacotes no tráfego de dados de um nó de rede. Esses pacotes podem ter como destinatário um endereço da rede interna (ie circular de fora da rede para dentro da rede) ou ter como destinatário um endereço fora da rede (ie circular de dentro da rede para fora da rede), isto relativamente à rede onde se encontra o *Firewall*.

Os objectivos da filtragem de pacotes podem ser variados. Ou porque interessa que um determinado servidor na rede receba todos os pacotes de um determinado tipo, ou porque determinado tipo de pacotes não deve poder entrar (ou sair) da rede, por questões de segurança, ou para não dar acesso a determinado tipo de serviços (*download* de filmes e MP3 que são grandes consumidores de recursos), etc.

De notar que ao impedir que determinados pacotes circulem na fronteira que um *Firewall* implementa, pode-se estar a barrar o acesso a informação que se encontra algures na rede, e que por questões de segurança se pretende preservar, por exemplo.

Para que um *firewall* possa funcionar correctamente há pelo menos três elementos essenciais, para que se possa cumprir uma regra de filtragem de pacotes.

i) conhecer um endereço ou uma rede de origem
ii) conhecer um endereço ou uma rede de destino
iii) definir uma acção a executar quando um pacote com determinadas características proveniente desta origem e com aquele destino for identificado.

Com estes elementos fundamentais e mais alguns, como é o caso entre outros do protocolo e da porta em uso, cria-se aquilo que se designa por regra.

As regras agrupam-se em tabelas. Por defeito o sistema possui 3 tabelas: INPUT (para as regras que dizem respeito aos pacotes que chegam ao sistema), OUTPUT (para as regras que dizem respeito aos pacotes de saiam do sistema), e FORWARD (para os pacotes que são encaminhados para outro sistema (routeamento).

É possível definir tabelas adicionais, o que ajuda a melhorar a estruturação das regras do *firewall*.

Uma regra deverá conter uma acção a fazer com os pacotes que cabem na definição da mesma. Acções possíveis são ACCEPT, REJECT ou DROP (entre outras),para aceitar rejeitar ou descartar pacotes.

Dedique algum tempo para perceber como pode construir regras no seu *Firewall*. Utilize os comandos:

iptables –h
man iptables

e analise as capacidades (e complexidade) do comando **iptables**.

Nas aulas teremos oportunidade para experimentar algumas situações concretas, onde poderá avaliar a sua compreensão deste tema.

6.5 Interfaces de rede

Para que as comunicações em rede possam funcionar correctamente e se possam executar as operações anteriormente referidas, é necessário que a máquina possua pelo menos uma interface de rede devidamente configurada.

O seu *driver* deverá estar instalado no *kernel* do sistema, e deverá poder ser configurado de maneira a que seja possível operar no contexto da rede em que a máquina se encontra.

Esta configuração é feita com o comando **ifconfig**.

Na sua utilização mais básica, podemos atribuir um endereço IP à máquina dentro da gama de IP's válidos para a rede onde ela se encontra.

Se por questões de política de atribuição de endereços da rede em que se encontra inserido não lhe é permitido definir um endereço IP, então poderá sempre recorrer aos serviços de DHCP algures disponíveis na rede, que irão fornecer indicações ao sistema para as configurações correctas sejam efectuadas.
Utilize neste contexto o comando **dhclient**.

Além destes,

ifconfig
dhclient

analise ainda os comandos:

ifup
ifdown

6.6 Monitorização de rede

Quando experimentamos problemas com as comunicações em rede, só uma monitorização da mesma poderá revelar aquilo que se está a passar de errado.

Dispomos de várias ferramentas que nos podem auxiliar na identificação do problema de uma forma directa ou indirecta.

Procure para o efeito explorar o comando **ping**, que de uma forma imediata, lhe dá uma perspectiva do tempo de resposta que encontra na rede.

Explore igualmente o comando **traceroute**. Este também lhe dá uma perspectiva do tempo de trânsito dos pacotes na rede, embora efectue uma análise em cada etapa (entre cada dois nós) do percurso. Caso os pacotes passem por vários *routers* até chegarem ao seu destino, saberemos os tempos envolvidos em cada trajecto.
Com este comando é possível perceber onde estão os pontos mais lentos da rede, é possível perceber se há erros de roteamento, entre outras situações.

Outros comandos, como é o caso do **netstat**, permitem-nos, entre outros aspectos, observar as portas que se encontram abertas ou fechadas num sistema, o que pode ser importante, por exemplo, para se perceber porque é que não se consegue aceder a um serviço, ou ainda o comando **route**, que permite analisar a forma como um sistema trata as mensagens que são destinadas à rede na forma de uma tabela de roteamento, e como as encaminha caso esteja a assegurar a função de um *router*.

Analise o funcionamento destes comandos e faça as suas anotações no seu *log book*