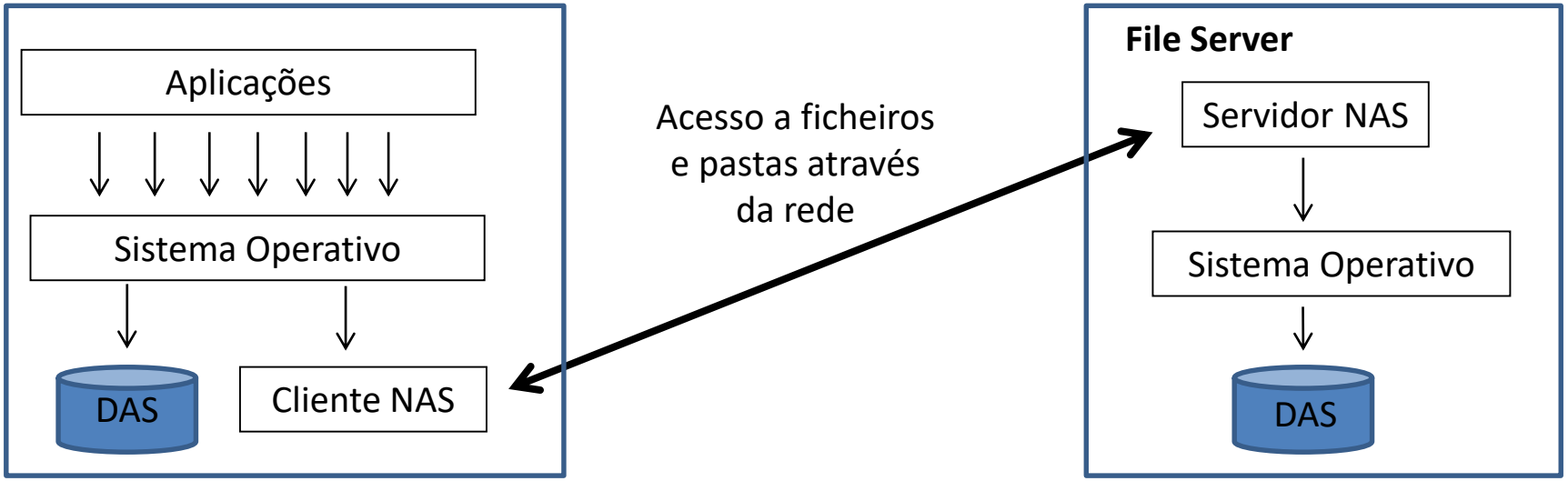
Administração de Sistemas (ASIST)

Aula Teórico Prática 06 - 2018/2019

Sistemas de ficheiros em rede (NAS) - SMB/CIFS e NFS
Filtragem de pacotes de rede – *Firewall*
Lightweight Directory Access Protocol (LDAP)
Linux – OpenLDAP

Network Attached Storage (NAS)

Os sistemas de ficheiros em rede, permitem partilhar de forma transparente através da rede diretórios (pastas). Para esse efeito são usados protocolos específicos segundo o modelo cliente-servidor, o servidor muitas vezes designado ***file server*** responde aos pedidos dos clientes através da rede facultando acesso aos seus ficheiros locais. O cliente está integrado no sistema operativo de tal forma que para utilizadores e aplicações não existe diferença aparente entre aceder a ficheiros locais ou a ficheiros remotos.



Este tipo de utilização de sistemas de ficheiros remotos através da rede designa-se NAS, por oposição a DAS (*Direct Attached Storage*) em que o sistema de ficheiros reside num disco local.

Protocolos NAS – SMB/CIFS

De entre outros destacam-se aqui dois protocolos de sistemas de ficheiros em rede: o **SMB/CIFS** e o **NFS**, originários respetivamente dos sistemas Windows e sistemas Unix. Atualmente quer servidores Windows quer servidores Linux podem usar ambos os protocolos.

SMB/CIFS (*Server Message Block/Common Internet File System*) refere-se a um protocolo desenvolvido ao longo do tempo pela Microsoft e que permite a partilha de sistemas de ficheiros através da rede. Efetivamente o protocolo é o SMB, CIFS refere-se ao dialeto mais recente suportado pelo SMB.

O SMB possui um mecanismo de negociação do dialeto, isto permite suportar clientes e servidores Windows de diferentes versões, nessa negociação os dois intervenientes determinam qual é o dialeto mais recente que é suportado por ambos. No SMB os dialetos são identificados por nomes, o mais antigo é o **PC NETWORK PROGRAM 1.0**, os mais recentes são o **DOS LANMAN 2.1** e o **NT LM 0.12**, este último corresponde ao CIFS.

O acesso a sistemas de ficheiros SMB/CIFS é orientado por sessão de utilizador, o utilizador autentica-se no servidor e obtém uma listas de partilhas disponibilizadas pelo servidor para esse utilizador. Além de pastas do sistema de ficheiros, também impressoras podem ser partilhadas. O acesso às partilhas é condicionado pelas permissões (ACLs) de acordo com o utilizador que se autenticou.

Claro que no contexto de um domínio Windows, depois de efetuar o login no domínio, o utilizador está desde logo autenticado em todos os servidores que são membros do domínio.

As partilhas de pastas via SMB/CIFS são identificadas através do formato UNC (*Universal Naming Convention*):

\\{NOME-SERVIDOR}\{NOME-PARTILHA}

Onde **{NOME-SERVIDOR}** representa o servidor que está a disponibilizar a partilha, podendo ser um nome simples NetBIOS, um nome DNS ou um endereço IP. **{NOME-PARTILHA}** é o nome que foi dado à partilha no servidor, normalmente as pastas são partilhadas usando o nome da pasta, no entanto podem ser definidas várias partilhas para uma mesma pasta, cada uma com um nome diferente.

Cada partilha tem uma ACL associada que define as permissões de acesso dos utilizadores e grupos. As ACL de partilhas só possuem três permissões: **Full Control**, **Change** e **Read**, as permissões definidas para a partilha não se sobrepõem às permissões no sistema de ficheiros NTFS.

Por omissão quando se cria uma partilha é dada a permissão **Read** ao grupo implícito **Everyone**, mas isto não significa que todos os utilizadores tenham acesso, tudo depende das permissões NTFS definidas na pasta que está a ser partilhada. É mesmo uma prática corrente atribuir nas partilhas a permissão **Full Control** ao grupo **Everyone**, deste modo todo o controlo é feito pelas ACLs do NTFS.

Domínios Windows - partilha da *home* e *profile*

As contas de utilizadores Windows definem os atributos ***Profile Path*** e ***Home Folder***. No contexto de um domínio estes atributos têm de corresponder a partilhas de rede disponibilizadas por servidores do domínio, só dessa forma estarão acessíveis em todos os postos de trabalho que efetuam login no domínio.

As pastas de trabalho pessoais dos utilizadores (***Home Folder***) deverão ser colocadas numa pasta partilhada por um dos servidores do domínio, por exemplo \\SERVER\HOMES, o atributo ***Home Folder*** de cada conta de utilizador poderá então assumir a forma **\\SERVER\HOMES\{username}**, onde **{username}** é o nome do utilizador (admitindo que cada pasta têm um nome igual ao nome do utilizador).

A partilha \\SERVER\HOMES pode ser definida com a permissão **Full Control** para **Everyone**, as permissões efetivas serão ditadas pelas ACLs do NTFS.

Para o ***Roaming Profile*** (***Profile Path***), estratégia idêntica deve ser implementada, possivelmente usando outro servidor do domínio.

Numa configuração mais simples pode ser usada para ***Profile Path*** uma subpasta de ***Home Folder***), por exemplo:

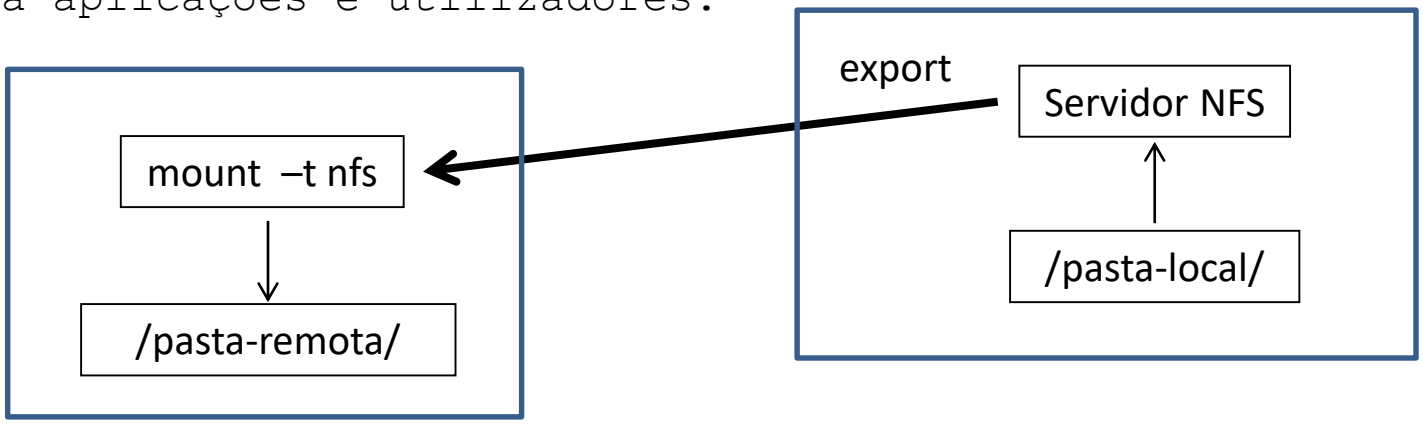
\\SERVER\HOMES\{username}\Profile\{username}

Protocolos NAS – NFS (Network File System)

O protocolo NFS é originário dos sistemas Unix, foi desenvolvido inicialmente pela *Sun Microsystems*. As implementações tradicionais NFSv2 e NFSv3 são totalmente vocacionadas para serem manuseadas exclusivamente pelos administradores dos sistemas, não existe suporte para autenticação de utilizadores.

No NFSv4 são suportadas novas formas de autenticação, incluindo a autenticação de utilizadores individuais.

Focalizando na utilização pelos administradores de sistemas, o NFS permite **exportar pastas locais** de uma máquina que atua como servidor NFS (o verbo exportar é razoavelmente equivalente a partilhar). Uma pasta exportada por NFS pode ser **montada por outra máquina** (cliente NFS) e desta forma fica integrada no sistema de ficheiros e disponível para aplicações e utilizadores.



Unix – NFSv2 e NFSv3 – NFS Server

Nos sistema Unix o servidor NFS é configurado através do ficheiro **/etc/exports**, neste ficheiro definem-se as pastas locais que são exportadas e em que condições.

No NFSv2 e NFSv3 o controlo de acesso às pastas exportadas é realizado através do endereço IP do cliente, assim no ficheiro **/etc/exports**, para cada pasta exportada são enumerados os endereços IP dos clientes aceites. O endereço IP pode ser especificado de várias formas, incluindo um endereço IP, uma rede IP (endereço da rede e prefixo), um nome que seja resolvido para um endereço IP e através de padrões, por exemplo * corresponde a qualquer endereço.

Para cada exportação podem ser definidas várias opções, incluindo leitura apenas (**ro**), leitura e escrita (**rw**). Por omissão as exportações usam sempre a opção **root_squash**, isto significa que o utilizador **root** da máquina cliente não terá sobre o sistema de ficheiros exportado as mesmas permissões que o utilizador **root** da máquina servidora possui.

Os restantes utilizadores terão exatamente as mesmas permissões tanto no servidor como no cliente, desde que os UID sejam os mesmos.

Exemplo de ficheiro **/etc/exports**:

```
/users/home         190.130.60.0/255.255.255.0(ro)
/users/backups      vsrv2(rw)   *(ro)
/usr/local/dump     190.130.60.10(rw)      190.130.60.123(rw)
```

Unix – NFSv2 e NFSv3 – Cliente NFS

Os clientes NFS autorizados podem montar os sistemas de ficheiros exportados pelo servidor usando o comando mount:

mount –t nfs {SERVER}:{PASTA} {PONTO-DE-MONTAGEM}

Onde **{SERVER}** representa o endereço IP ou nome DNS do servidor NFS, **{PASTA}** representa o nome da pasta exportada (tal como consta no ficheiro /etc/exports do servidor) e **{PONTO-DE-MONTAGEM}** representa uma pasta local vazia onde será disponibilizada a pasta remota.

Para tornar a configuração permanente, a montagem deve ser adicionada ao ficheiro **/etc/fstab** para que seja realizada no arranque da máquina.

Todos os atributos do sistema de ficheiros são exportados, incluindo o UID (owner), GID e permissões associados a cada objeto.

Consequentemente as permissões dos utilizadores da máquina cliente sobre o sistema de ficheiros montado são exatamente as mesmas que possuem no servidor, só é necessário garantir que o UID e GID de cada utilizador e grupo é exatamente o mesmo em todas as máquinas.

Garantir que o UID e GID é o mesmo em todas as máquinas é facilmente conseguido se todas as máquinas utilizarem um mesmo repositório de contas de utilizadores e grupos centralizado, por exemplo LDAP.

Autenticação por endereço IP – NFSv2 e NFSv3

A autenticação por endereço IP só pode ser considerada segura se for aplicada com algumas precauções:

- **Tem de haver confiança total no administrador do nó cliente.** Pelo facto de serem usados números de porto reservados (<1024), tanto o cliente como o servidor sabem que estão a dialogar com uma aplicação sob o controlo do administrador do nó. Particularmente quando o servidor recebe um pedido de um cliente verifica que provém de um endereço IP autorizado e de um número de porto reservado.

- **Tem de haver confiança no funcionamento da rede.** O sistema é totalmente vulnerável a ataques de *IP Spoofing*, consequentemente só deve ser usado numa rede a que os utilizadores não têm acesso, tipicamente uma rede onde apenas existem servidores, uma DMZ (*DeMilitarized zone*).

- **Tem de haver confiança no funcionamento da resolução de nomes.** No caso de se usarem nomes de máquinas em lugar de endereços, o sistema poderia ser atacado através do DNS. Se o servidor DNS for comprometido poderá dar respostas falsas e desse modo levar o servidor NFS a permitir os endereços errados. Como medida de segurança, no contexto do NFS, é aconselhável usar sempre endereços IP e não nomes DNS de máquinas.

Utilização do NFS para partilha das *homes*

Num ambiente em que existe um conjunto de servidores Linux, a utilização de um repositório de contas de utilizadores e grupos centralizado, por exemplo, numa base de dados acessível por LDAP, permite que os utilizadores e grupos sejam iguais em todos os servidores.

Consequentemente os utilizadores podem efetuar o login em qualquer dos servidores, o NFSv2/NFSv3 pode então ser usado para igualmente disponibilizar as ***home directories*** em todos os servidores.

As ***home directories*** ficarão alojadas numa pasta de um servidor que é exportada por NFS para todos os servidores onde as *home directories* são necessárias. A pasta contendo as *home directories* deverá ser exportada em ***read-write***. Como o NFS mantém todas as propriedades dos objetos, nomeadamente UID/GID associados e permissões, os utilizadores terão em qualquer dos servidores as permissões corretas sobre a respetiva *home directory*.

Nos servidores que recebem a pasta exportada é necessário ter o cuidado de montar a pasta exportada num local que permita que o atributo *home directory* da conta de utilizador corresponda à efetiva localização da pasta.

Eventualmente se não corresponderem também será simples resolver o problema através de uma ligação simbólica.

Filtragem de pacotes - firewall

Designam-se ***firewalls*** mecanismos integrados nos nós de rede que se destinam a condicionar a passagem de tráfego de rede.

Um *firewall* num nó intermédio como um ***router*** permite controlar o tráfego que é transferido entre diferentes redes. Um *firewall* num nó final como um servidor ou um posto de trabalho permite condicionar o tráfego que entra (***inbound***) e sai (***outbound***) no nó.

No caso de um nó final, todas as atenções se concentram no tráfego ***inbound*** pois assume-se que as ameaças provêm da rede e não do interior do nó.

Existem vários formas de analisar o tráfego de rede correspondendo a diferentes designações para o tipo de firewall:

- Os mais simples procedem a uma análise individual de cada pacote e são normalmente designados ***stateless*** ou ***packet filters***.
- Por oposição um firewall designa-se ***stateful*** se procede a uma análise das transações completas ao nível dos protocolos de transporte ***connection-oriented*** e dos protocolos de aplicação.

Note-se que a proteção contra ataques do tipo DoS (***Denial of Service***) só pode ser assegurada através de ***firewalls stateful***.

Windows Server - Windows Firewall with Advanced Security

As regras *stateless* dividem-se em ***Inbound Rules*** e ***Outbound Rules***, respetivamente aplicadas ao tráfego que provém da rede e tráfego que o nó pretende emitir.

Os pacotes que chegam da rede ou que se pretendem emitir para a rede são confrontados com as listas de regras, respetivamente *Inbound* e *Outbound*.

Cada regra define critérios relativos a quais os pacotes que lhe correspondem e a ação a desencadear em caso de correspondência: ***Allow***, ***Block*** e ***Allow if secure***.

No último caso, os critérios para considerar ***secure*** podem ser definidos para cada regra.

O tráfego **inbound** só é permitido se corresponder a uma regra ***Allow*** e não corresponder a nenhuma regra ***Block***. (nas ***Inbound Rules***)

O tráfego ***outbound*** só é bloqueado se corresponder a uma regra ***Block***. (nas ***Outbound Rules***).

Como se conclui a aplicação das regras **inbound** é muito mais restritiva dos que a aplicação das regras ***outbound***.

No caso do **inbound**, não havendo nenhuma correspondência o pacote é **bloqueado**. No caso do ***outbound***, não havendo nenhuma correspondência o pacote é **permitido**.

Para cada regra define-se os critérios de correspondência com pacotes, podem ser baseados em:

- **Programs and Services** - aplicações locais às quais os pacotes se destinam (*inbound*) ou nas quais os pacotes têm origem (*outbound*).
- **Protocols and Ports** – protocolo (TCP, UDP, ICMPv4, …) e, quando aplicável, números de porto de origem e destino (TCP e UDP) ou tipo de mensagem ICMP.
- **Scope** – endereços de nó de origem e destino do pacote (IPv4 e/ou IPv6).
- **Local Principals** – utilizadores locais (a quem pertencem as aplicações que vão receber o pacote ou emitiram o pacote).
- **Remote Users** – utilizadores remotos (do *Active Directory*).
- **Remote Computers** – máquinas remotas (do *Active Directory*).

Também é possível condicionar a aplicação das regra ao contexto de rede em que a máquina se encontra (***Profile***):

- **Domain** – se máquina está ligada a uma rede que pertence ao domínio.
- **Private** – se a máquina está ligada a uma rede que foi classificada como privada.
- **Public** – se a máquina está ligada a uma rede que foi classificada como pública.

Linux – Netfilter e comandos iptables

O **Netfilter** é um sistema de controlo de tráfego integrado no *kernel* Linux, permite entre outras funcionalidades definir regras relativamente aos pacotes de rede que são permitidos.

Para gerir o Netfilter o administrador usa os comandos **iptables** e **ip6tables**, respetivamente para regras relativas ao IPv4 ou IPv6.

Através do iptables podem ser geridas 5 tabelas: **filter**, **nat**, **mangle**, **raw** e **security**.

A tabela **filter** é usada para definir critérios de permissão (**ACCEPT**) ou não permissão (**DROP**) de pacote. A tabela **nat** é usada para definir regras de manipulação de endereços (***Network Address Translation***), de origem (SNAT) e de destino (DNAT)

Ao utilizar o comando **iptables**, podemos indicar a tabela que pretendemos manipular através da opção **-t**, por omissão a tabela usada é **filter**.

Em cada tabela existem cadeias de regras (***chains***). Cada tabela possui algumas cadeias pré-definidas (*built-in*), mas podem ser criadas outras cadeias. Cada cadeia pré-definida aplica-se a pacotes em determinadas situações.

Em cada cadeia, as regras (rules) são numeradas a partir de 1 (primeira regra). Cada pacote aplicável à cadeia é confrontado sequencialmente com cada regra da cadeia.

Table	Built-in chains	Objetivos
filter	INPUT	Sequência de regras ACCEPT/DROP aplicadas aos pacotes quando chegam da rede e se destinam ao nó.
	FORWARD	Sequência de regras ACCEPT/DROP aplicadas aos pacotes quando chegam da rede, mas não se destinam ao nó. Utilizada quando o nó opera como **router**.
	OUTPUT	Sequência de regras ACCEPT/DROP aplicadas aos pacotes que vão ser emitidos pelo nó e têm origem no nó.
nat	PREROUTING	Alterações de endereços a aplicar a pacotes que chegam ao nó, (sejam ao não destinados ao nó), antes do pacote ser encaminhado.
	OUTPUT	Alterações de endereços a aplicar a pacotes emitidos pelo nó (têm origem no nó) antes de serem encaminhados.
	POSTROUTING	Alterações de endereços a aplicar a pacotes que vão ser emitidos pelo nó (tenham ou não origem no nó), após o pacote ser encaminhado.

Cada regra de uma cadeia define **critérios de correspondência** com pacotes e a **ação a realizar** (**target**) em caso de correspondência. Não correspondendo com a regra o pacote será confrontado com a regra seguinte da cadeia.

Se um pacote atinge o fim de uma cadeia, será executada sobre ele a ação definida na **política da cadeia** (apenas para cadeias *built-in*).

iptables - targets

Cada regra define a ação a realizar em caso de correspondência de um pacote através das opções **-j target** ou **-g chain.**

Ação	Descrição da ação sobre o pacote
-j ACCEPT	Deixar passar o pacote, termina o processamento da cadeia para o pacote.
-j DROP	Bloqueia (destrói) o pacote, termina o processamento da cadeia para o pacote.
-g {chain}	Processa a cadeia identificada por {chain}. Para além das built-in é possível criar novas cadeias de regras. Usando a opção –g, o pacote será confrontado com as regras da cadeia designada.
-j RETURN	Termina o processamento da cadeia atual e regressa à cadeia de origem para continuar o processamento das regras seguintes. Se a cadeia atual é built-in aplica ao pacote a politica da cadeia.
-j SNAT	Alteração do endereço de origem (IP e/ou porto) (através da opção --to-source). Apenas é válido na cadeia POSTROUTING da tabela nat. Termina o processamento da cadeia para o pacote.
-j DNAT	Alteração do endereço de destino (IP e/ou porto) (através da opção --to-destination). Apenas é válido nas cadeias PREROUTING e OUTPUT da tabela nat. Termina o processamento da cadeia.
-j LOG	Regista o pacote nos logs do sistema. Não termina o processamento da cadeia, continua a processar o pacote através da regra seguinte da cadeia.

iptables - critérios de correspondência

Cada regra define **critérios de correspondência** com pacotes, os critérios são especificados através de opções. Sendo definidos vários critérios, a **regra apenas será aplicada a pacotes para quais todos os critérios se verificam.** Os critérios podem ser negados se a opção for precedida pelo símbolo !.

Critério	Descrição do critério
-i {iface}	O pacote foi recebido através da interface de rede {iface}.
-o {iface}	O pacote vai ser emitido através da interface de rede {iface}.
-p {proto}	O pacote transporta dados do protocolo designado {proto}. O valor de {proto} pode ser por exemplo tcp, udp, icmp ou icmpv6. O valor all representa qualquer protocolo, é equivalente a não definir este critério.
-s {address}[/{mask}]	O pacote tem como endereço de origem os endereços especificados, se for omitida /{mask} o endereço é interpretado com sendo um único nó, ou seja mask 32 ou 128 bits respetivamente para IPv4 e IPv6.
-d {address}[/{mask}]	Idêntica à opção –s, mas especificando o endereço de destino do pacote.
--dport {port}[:port2]	O número de porto de destino do pacote é {port}, ou está dentro do intervalo [{port}, {port2}] (apenas válido para –p udp ou –p tcp).
--sport {port}[:port2]	O número de porto de origem do pacote é {port}, ou está dentro do intervalo [{port}, {port2}] (apenas válido para –p udp ou –p tcp).

iptables – definição das cadeias de regras

Comando	Descrição do comando
-N {chain}	Cria uma nova cadeia.
-X {chain}	Elimina uma cadeia anteriormente definida (não aplicável para cadeias build-in).
-F [{chain}]	Elimina todas a regras da cadeia, se a cadeia for omitida elimina todas as regras de todas as cadeias da tabela.
-P {built-in-chain} {target}	Define a política da cadeia, apenas válido para cadeias built-in, os targets válidos são ACCEPT e DROP.
-A {chain} {rulespec}	Acrescenta uma regra (append) à cadeia designada. {rulespec} é uma definição de regra, incluindo critérios e target.
-I {chain} [{num}] {rulespec}	Insere na cadeia uma nova regra na linha {num}, se {num} não é especificado insere na primeira posição.
-R {chain} {num} {rulespec}	Substitui a regra que se encontra na posição {num} da cadeia.
-D {chain} {num}	Elimina a regra que se encontra na posição {num} da cadeia.
-L [{chain}]	Apresenta a lista das regras na cadeia designada ou de todas as cadeias da tabela. A opção **–n** apresenta todos os endereços e portos na forma numérica. A opção **–v** apresenta mais detalhes. A opção **--line-numbers** apresenta a numeração das regras.

iptables - módulos

A opção –m permite carregar módulos para suportar funcionalidades adicionais. Por exemplo o módulo **state** (-m state) permite definir o critério de correspondência **--state {estados}**, onde estados é uma lista separada por virgulas de valores tais como: ESTABLISHED, NEW e RELATED. O critério ESTABLISHED aplica-se a tráfego TCP relativo a conexões já estabelecidas, NEW a novas conexões TCP. O critério RELATED aplica-se a tráfego UDP de resposta, ou seja pacotes que correspondem a respostas de pedidos UDP anteriormente realizados.

Linux – *routing*

Por omissão, o *kernel* Linux rejeita pacotes provenientes das redes que não lhe sejam destinados (não encaminha tráfego entre redes). Esse comportamento pode ser alterado modificando as variáveis de sistema **net.ipv4.ip_forward** e **net.ipv6.conf.all.forwarding** para o valor 1, respetivamente para IPv4 e IPv6. O nó passará a operar como *router*.

Para tornar estas configurações permanentes (aplicadas sempre que o sistema arranca) devem ser definidas no ficheiro de configuração **/etc/sysctl.conf**.

LDAP (*Lightweight Directory Access Protocol*)

O protocolo LDAP foi concebido para permitir um acesso expedito a serviços de informação através da rede, tipicamente consultas e pesquisa de informação.

Sob o ponto de vista do LDAP a informação é guardada em objetos e organizada numa estrutura em árvore (***DIT - Directory Information Tree***). Cada objeto é a instanciação de uma classe, os objetos contêm atributos onde os dados propriamente ditos são armazenados.

A classe (***LDAP objectClass***) define os atributos dos objetos, tais como:

- Quais são os atributos e o seu nome.
- Que tipo de dados um atributo pode guardar (que valores pode assumir).
- Se o atributo é obrigatório ou opcional.
- Se o atributo pode ser repetido várias vezes no objeto ou tem de ser único.

As classes de objetos são normalizadas e não devem ser definidas arbitrariamente, as classes são definidas através do ***schema***.

Existe uma grande variedade de classes disponíveis para os mais diversos tipos de aplicação.

LDAP - classes e objetos

A definição de classes (no *schema*) recorre à herança, ou seja pode definir-se uma nova classe a partir de uma classe existente e acrescentar novos atributos. Existem 3 tipos de classe:

ABSTRACT – é uma classe de topo definida com o objetivo de ser herdade por classes concretas. Não se podem criar objetos com este tipo de classes.

STRUCTURAL – é uma classe com a qual se podem criar objetos, um objeto tem de pertencer a uma classe estrutural, mas não pode pertencer a mais do que uma.

AUXILIARY – é uma classe com o propósito de acrescentar atributos a um objeto que já pertence a uma classe estrutural.

Um objeto tem de pertencer a uma e uma só classe estrutural, mas pode pertencer a várias classes auxiliares.

Exemplo de parte de um *schema*:

```
objectclass ( 2.5.6.2 NAME 'country' SUP top STRUCTURAL
  DESC '2 character iso 3611 assigned country code'
  MUST c
  MAY ( searchGuide $ description ) )
```

LDAP – DN (*Distinguished Name*) e root DN

Cada objeto na DIT é identificado de forma única através do DN. No seio do ramo da DIT em que se encontra um objeto é identificado de forma única através do valor de um dos seus atributos esse é o RDN (Relative Distinguished Name) do objeto. O DN é constituído pelo RDN seguido do DN do ramo (DN do objeto pai). Os vários elementos do DN e RDN são separados por uma virgula.

A raiz da DIT (topo da hierarquia) tem ela própria um DN que é definido pelo administrador (root-DN). Qualquer objeto de uma DIT terá necessariamente incluído no seu DN o root-DN da DIT.

Segundo as normas em vigor, o root-DN deve corresponder ao nome DNS do domínio local separado em componentes através do atributo dc (*Domain Component*):

dc=sub-domínio,dc=sub-domínio,…,dc=domínio-de-topo

Por exemplo para o domínio DNS dei.isep.ipp.pt, o root-DN da DIT deve ser: **dc=dei,dc=isep,dc=ipp,dc=pt**

O objeto com **dn: cn=user1,ou=utilizadores,dc=dei,dc=isep,dc=ipp,dc=pt** fica perfeitamente identificado, tem como RDN **cn=user1** e está localizado no ramo **ou=utilizadores,dc=dei,dc=isep,dc=ipp,dc=pt.**

Linux – Servidor OpenLDAP (http://www.openldap.org)

Um dos servidores LDAP mais usado em Linux é o OpenLDAP. Tradicionalmente a configuração é realizada através do ficheiro **slapd.conf**, mas em distribuições recentes como o Ubuntu atual, a configuração é definida numa DIT do próprio servidor com root-DN **cn=config**.

Alguns dos elementos base de configuração são:

- Os **schema** a suportar (classes de objetos).
- **Backend** – a base de dados onde os objetos são armazenados, o mais comum é bdb (*Oracle Berkeley DB*), mas também pode usar SQL.
- **Suffix** – refere-se ao root-DN da DIT.
- **Conta de administração** – identifica um objeto na DIT que corresponderá ao administrador LDAP, o objeto deve conter o atributo userPassword.
- **Atributos a indexar** - para agilizar a pesquisa na árvore, os atributos mais relevantes devem ser indexados. Desde logo o atributo objectClass. Para contas de utilizadores outros atributos como cn, uid, uidNumber e gidNumber devem também ser indexados.

Uma vez definida as configurações base, a conta de administração pode ser usada para gerir os objetos a partir da root-DN definida e, se aplicável, gerir a configuração residente na DIT **cn=config**.

LINUX - LDAP – ficheiros LDIF

Estabelecida a conta de administração, o LDAP pode então ser usado para gerir os objetos nas DIT, para esse efeito podem ser usados os comandos:

ldapadd – adicionar um objeto.

ldapdelete – eliminar um objeto.

ldapmodify – adicionar, eliminar ou alterar um objeto (atributos), incluindo alterar o nome (DN). Note-se que alterar o DN pode incluir mover o objeto de um ramo para outro (moddn) ou não (modrdn).

Estes comandos aceitam como argumento ficheiros em formato **LDIF** (***LDAP Data Interchange Format***) com instruções das ações a realizar usando uma sintaxe específica, por exemplo:

```
dn: cn=user1,dc=example,dc=com
        changetype: modify
        replace: mail
        mail: modme@example.com
        -
        add: title
        title: Grand Poobah
        -
```

Se um ficheiro com este conteúdo for fornecido ao comando **ldapmodify** iria alterar o conteúdo do atributo ***mail*** e acrescentar o atributo ***title*** no objeto com DN **cn=user1,dc=example,dc=com.**
Cada ação definida deve terminar com uma linha constituída pelo símbolo **-.**