Administração de Sistemas (ASIST)

Aula Teórico Prática 02

Utilizadores e grupos.
Administração centralizada.

Baseado em: A. Moreira, 2018/2019, Aulas teórico-práticas de ASIST

Utilizadores, grupos e permissões

O conceito de utilizador é comum a todos os sistemas operativos *multiuser*, é fundamental para distinguir diferentes atores perante o sistema. Cada utilizador tem associado a si um conjunto de **atributos** que constituem o que vulgarmente se designa **conta de utilizador** (*user account*).

As contas de utilizadores são armazenadas pelo sistema operativo em bases de dados apropriadas. Diferentes sistemas operativos utilizam tipos de contas de utilizador diferentes com diferentes tipos de atributos, consequentemente este é um aspeto de difícil integração.

Entre os atributos que são comuns a todos os sistemas operativos encontra-se o ***login name*** através do qual o utilizador se identifica perante o sistema.

A cada utilizador o administrador pode atribuir permissões (o que lhe é permitido fazer). O conceito de grupo de utilizadores (*group account*) vem facilitar esta tarefa uma vez que as permissões atribuídas a um grupo têm efeito sobre todos os seus membros.

As permissões podem ser definidas sobre a forma de direitos (*user rights*) associadas à conta de utilizador ou de grupo, ou sob a forma de uma **ACL** (*Access Control List*) associada a um objeto.

Sistemas Unix – Utilizadores

Nos sistemas Unix a base de dados fundamental de utilizadores é o ficheiro de texto **/etc/passwd**. Outras bases de dados podem ser usadas em complemento a esta.

```
root:x:0:0:root:/root:/bin/bash
nobody:x:65534:65534:Nobody:/:/bin/sh
www:x:98:98:Gestor WWW:/users/home/www:/usr/bin/tcsh
i977805:x:2176:102:Joaquim Cardoso Morais:/users/home/i977805:/bin/csh
artur:$1$s2b9pyc6$46nIOl8G1fGrymmvsJejG/:1203:98:Artur:/users/home/artur:/bin/bash
```

A imagem acima ilustra um exemplo do conteúdo deste ficheiro, neste caso com 5 contas, começando pelo utilizador **root** (administrador do sistema, possui todas as permissões). Cada linha do ficheiro corresponde a uma conta de utilizador e contém **7** atributos separados pelo símbolo de dois pontos:

- *Login-name* – nome único que identifica o utilizador no sistema.
- *Hash* da *password*. O valor **x** indica que o *hash* se encontra no ficheiro **/etc/shadow**
- UID (*User IDentifier*) – número único de 16 bits que identifica o utilizador.
- GID (*Group IDentifier*) primário – número de 16 bits que indica o GID do grupo primário deste utilizador.
- Nome longo ou descrição do utilizador.
- Pasta inicial (*home*). Pasta corrente após o *login*.
- Programa inicial. Tipicamente uma *shell* executada após o *login* (se não estiver preenchido, aplica-se a *shell* por omissão).

Sistemas Unix – Autenticação de utilizadores

O *hash* da password guardado no ficheiro **/etc/passwd** é usado para autenticação, quando o utilizador fornece a *password* o sistema aplica-lhe a função *hash* e compara o resultado com o que está no **/etc/passwd**; sendo iguais a autenticação é válida.

O sistema operativo exige que o ficheiro **/etc/passwd** seja legível por todos os utilizadores, consequentemente concluiu-se que não é o local ideal para guardar os *hash* das *passwords* dos utilizadores.

Note-se que uma função *hash* não é reversível, por isso não é possível obter a *password* a partir do *hash*, mas o facto de haver acesso a essa informação abre a possibilidade de ataques de força bruta tentando várias *passwords* aleatórias e vendo se por acaso após aplicar a função *hash* a alguma delas o resultado é igual ao que está no **/etc/passwd**.

Para evitar esta vulnerabilidade os *hash* das passwords no **/etc/passwd** podem ser substituídos pelo símbolo **x** que significa que o *hash* se encontra no ficheiro auxiliar **/etc/shadow**, não sendo possível aos utilizadores lerem este ficheiro auxiliar.

De notar que a colocação da *hash* no ficheiro **/etc/shadow** não impede os ataques de força bruta, apenas os dificulta pois não é possível a um utilizador comparar a *hash* gerada com aquela que está armazenada.

Sistemas Unix – Grupos de utilizadores

Nos sistemas Unix, as contas de grupos de utilizadores são armazenadas no ficheiro de texto **/etc/group**, outras bases de dados podem ser usadas em complemento a esta.

O formato é idêntico ao do ficheiro **/etc/passwd**, na figura ao lado é apresentado um exemplo. Os atributos de cada grupo são o nome do grupo, o campo seguinte (**x**) não é usado atualmente, segue-se o **GID** (*Group IDentifier*). Quer o nome do grupo quer o GID têm de ser únicos.

```
root:x:0:
bin:x:1:
nogroup:x:65534:
sshd:x:77:
users:x:100:
inf:x:102:
profs:x:98:
dom_users:x:1003:artur,i977805
```

A primeira linha do exemplo define o grupo **root**. Embora nomes de grupos e nomes de utilizadores sejam totalmente independentes, por omissão é criado para cada utilizador um grupo com o mesmo nome que é o grupo primário desse utilizador.

O último atributo é uma lista separada por virgulas de nomes de utilizadores que são membros do grupo (como grupo secundário) para além do respetivo grupo primário.

Um utilizador pode pertencer a um grupo por duas vias: porque é o seu grupo primário (definido no ficheiro /etc/passwd) ou porque está na lista de membros do grupo (no ficheiro /etc/group).

Linux – Gestão de utilizadores e grupos

A gestão de utilizadores e grupos locais (definidos nos ficheiros **/etc/passwd**, **/etc/group** e **/etc/shadow**) deve ser realizada através de um conjunto de comandos padrão do Linux existentes na maioria das distribuições: **useradd ; usermod ; userdel ; groupadd ; groupmod ; groupdel ; chfn ; chsh ; passwd; pwconv.**

Estes comandos têm a vantagem de terem uma solidez comprovada e realizarem algumas validações e operações acessórias.

Uma alternativa que deve ser utilizada com cautela consiste na manipulação direta dos ficheiros, ou seja usar um editor de texto para alterar os ficheiros **/etc/passwd** e **/etc/group**. Neste caso há necessidade de evitar qualquer tipo de incoerência ou sobreposição de identificadores. Estes ficheiros são lidos sequencialmente, se houver uma duplicação de identificadores, apenas o primeiro será válido.

Note-se que o utilizador **root** (UID=0) e grupo **root** (GID=0) são fundamentais para o funcionamento do sistema, sem elas o sistema nem sequer poderá arrancar.

Sistemas Windows – Utilizadores e Grupos

No Windows Server só existem contas de utilizadores locais e grupos locais se o servidor não estiver integrado em nenhum domínio Windows, ou seja estiver a operar como um *Standalone Server*.

Num sistema Windows *standalone*, as contas de utilizador estão armazenadas numa base de dados do sistema operativo designada SAM (*Security Account Manager*). Quando um sistema Windows é integrado num domínio deixa de utilizar a SAM local e passa a usar a base de dados *Active Directory* do domínio.

A quantidade de atributos associada às contas de utilizador e grupos Windows é bastante extensa, podendo mesmo ser expandida no caso do *Active Directory*. Uma diferença que desde já se pode destacar relativamente ao Unix é que ao contrário daquele, nos sistemas Windows **um grupo pode ser membro de outro grupo**. Uma outra diferença é que nos sistemas Windows contas de grupo e de utilizadores são armazenadas conjuntamente e em consequência não é possível ter um nome de utilizador igual a um nome de grupo.

Quando o Windows Server é instalado são criadas automaticamente diversas contas de utilizador e grupo fundamentais para o funcionamento do sistema. O nome do utilizador que possui direitos totais de administração é ***Administrator*** e pertence ao grupo ***Administrators*** (se a versão do Windows for a inglesa).

Nomes de Utilizadores e de Grupos

Nos sistemas **Windows** os nomes de utilizadores e grupos podem ter até 64 caracteres. Alguns caracteres são proibidos, mas podem ser usados **espaços e pontos**, a Microsoft desaconselha a utilização de espaços no caso de nomes de utilizadores. Os nomes de utilizadores e de grupos podem ter maiúsculas e minúsculas, mas **não são *case sensitive***.

Nos sistemas **Linux** as restrições são maiores, não devido ao sistema operativo propriamente dito mas devido a um vasto conjunto de *software open source* com o qual é necessário manter a compatibilidade. As recomendações são as seguintes:

- Não conter letras maiúsculas.
- O nome tem de começar por uma letra minúscula ou um *underscore*.
- Os caracteres seguintes também podem ser dígitos numéricos ou o sinal menos. Adicionalmente o último caractere também pode ser o dólar. Não são permitidos outros caracteres, incluindo o espaço.
- O nome não deve exceder os 31 carateres de comprimento.

Centralização de contas

Num ambiente de rede de trabalho típico, os vários utilizadores autorizados devem ter acesso a um conjunto de servidores e outros recursos. Não é obviamente praticável manter localmente em cada um deles uma base de dados de utilizadores e grupos independente.

A solução é ter uma única base de dados de contas, central, e que seja acessível a todos os equipamentos através da rede usando protocolos de aplicação apropriados. Dessa forma, qualquer alteração na base de dados central de contas tem efeito imediato sobre todos os equipamentos que a estão a usar.

Para os utilizadores é muito mais cómodo usar apenas um único conjunto de credenciais e não um diferente para cada recurso a que acede.

Com a centralização das contas obtém-se igualmente um **sistema de autenticação centralizada** dos utilizadores. Embora as credenciais sejam iguais em todos os recursos/serviços, o utilizador tem de se autenticar em cada um deles.

Um **sistema de autenticação *single sign-on* (SSO)**, é ainda mais cómodo para os utilizadores. Como o nome indica o utilizador apenas necessita de se autenticar uma vez, após isso tem acesso a todos os recursos. Este é o tipo de autenticação que temos nos domínios Windows da Microsoft.

Domínios Windows da Microsoft

Os domínios Windows surgiram com o Windows NT Server, trata-se de um sistema de autenticação centralizada ***single sign-on***.

O funcionamento do domínio é centralizado numa máquina Windows Server com a função ***Domain Controller*** (**DC**). O DC contém a base de dados com as contas de utilizadores, grupos e outros constituintes do domínio. Desde o Windows 2000 Server, as base de dados dos DC são do tipo *Active Directory* (**AD**), utilizando o *standard* **LDAP** (*Lightweight Directory Access Protocol*).

Antes da introdução do AD, cada domínio possuía um DC designado **PDC** (*Primary Domain Controller*) que continha a base de dados do domínio. Num domínio só pode existir um PDC, mas podem existir outros DC que contêm cópias de leitura apenas da base de dados.

Com a introdução do AD o conceito de PDC desaparece embora continue a ser emulado para compatibilidade com sistemas mais antigos. Um domínio AD pode conter vários DC, as bases de dados e funções de controlo necessárias (*Active Directory Domain Services* – **AD DS**) são distribuídas pelos vários DC existentes, alguns deles vão ter funções especais (***operations masters***).

Estrutura lógica do Active Directory - Tree

O ***Active Directory*** (**AD**) não se limita a domínios, existe uma infraestrutura lógica que integra os domínios.

Os nomes de domínio AD correspondem a nomes qualificados de domínios DNS. Uma super estrutura lógica de domínios que pode ser criada é a árvore de domínios (***Domain Tree***).

Uma árvore corresponde a um **conjunto de domínios DNS contíguos**, ou seja um domínio DNS e eventualmente subdomínios DNS. O domínio AD correspondente ao domínio DNS base (mais próximo da raiz) que pertence à árvore designa-se ***tree root domain***.

Um domínio AD pertence sempre a uma árvore, se for o único domínio da árvore é também o ***tree root domain***.

Quando um novo domínio é criado, por *promoção* de um Windows Server de ***Standalone Server*** a ***Domain Controller***, o novo domínio ou é integrado numa árvore já existente, se o nome DNS corresponde a um subdomínio de outro que já pertence a uma árvore, ou então será criada uma nova árvore, tendo o novo domínio como raiz.

Estrutura lógica do Active Directory - Forest

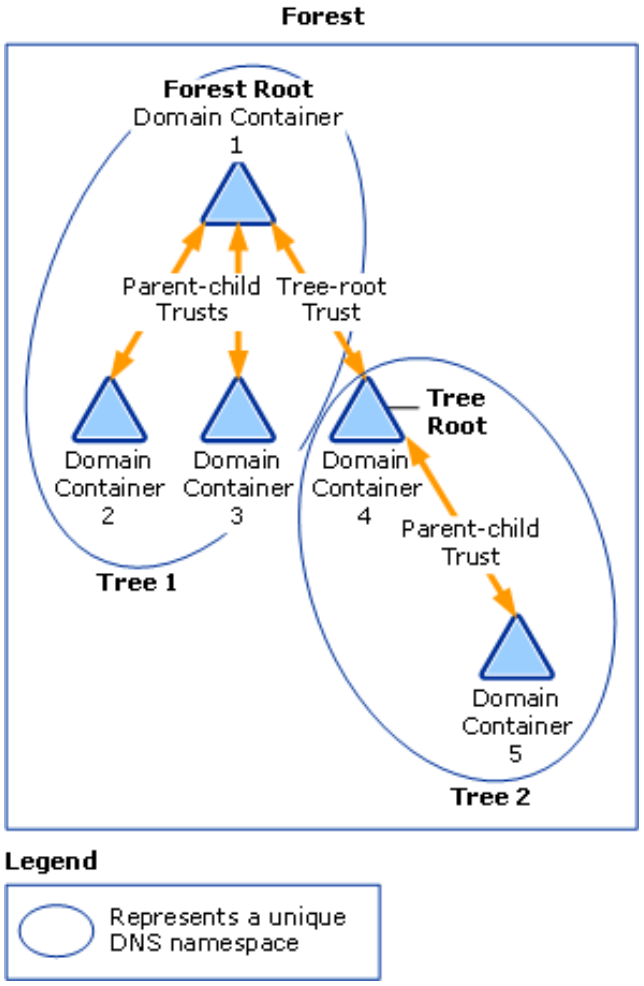
Uma floresta é um conjunto de árvores, e consequentemente um conjunto de domínios, e **ao contrário do que acontece com os domínios de uma árvore, pode não existir qualquer relação entre os nomes DNS das árvores que integram uma floresta.**

Quando um novo domínio é criado tem de ser integrado numa floresta já existente ou terá de ser criada uma nova floresta. O primeiro domínio a ser adicionado a uma floresta passa a ser o ***forest root domain*** e a floresta passa a ser identificada através do nome DNS desse domínio.

Uma floresta estabelece uma relação de confiança entre todos os seus domínios.

Dito de forma simples isto significa que um utilizador autenticado num dos domínios da floresta terá acesso aos recursos de outros domínios da floresta.

A fronteira da floresta no seu conjunto constitui uma *security boundary*. A floresta também pode ser vista como o nível mais elevado da organização lógica do AD.



Fonte: https://technet.microsoft.com

Active Directory Federation Services (AD FS)

Trata-se de um sistema de autenticação ***single sign-on*** (**SSO**) que permite ultrapassar os limites de confiança definidos pela floresta.

A federação não deve no entanto ser interpretada como sendo um nível lógico do AD acima das florestas.

Foi pensado para permitir a integração da autenticação entre organizações diferentes (***federation partners***) que desta forma constituirão uma **federação.**

Estes serviços estão vocacionados para serem usados diretamente por aplicações tais como serviços Web e utilizam a internet de forma segura para aceder a servidores de autenticação específicos designados ***AD FS servers.***

Os utilizadores de organizações que fazem parte da mesma federação (***federation partners***) necessitam de se autenticar apenas uma vez, após isso têm acesso às aplicações (eventualmente Web) disponibilizadas pelos ***federation partners*** sem necessidade de autenticações adicionais.

Active Directory Domain Controllers

A estrutura lógica do *Active Directory* é suportada por um conjunto de máquinas Windows Server designadas *Domain Controllers* (**DC**), ou seja disponibilizam os *Active Directory Domain Services* (**AD DS**).

Cada domínio AD tem de possuir pelo menos um DC, a Microsoft aconselha um mínimo de dois para garantir tolerância a falhas.

Uma máquina Windows Server pode encontrar-se em uma de 3 situações:

- Após a instalação inicial, não está integrado em nenhum domínio e designa-se ***Standalone Server***.
- Pode ser integrado num domínio sem ser um *Domain Controller*, neste caso designa-se ***Domain Member***, tal como os postos de trabalho.
- Pode ser promovido a ***Domain Controller***, eventualmente criando um novo domínio, uma nova árvore e uma nova floresta.

Dependendo do contexto em que é promovido a *Domain Controller* pode exercer ainda funções especiais no *Active Directory*, nomeadamente ***Global Catalogue Server*** e funções ***Operations Master***.

Em cada floresta existe um *global catalogue*, trata-se de uma base de dados que contém réplicas das partes mais importantes de cada uma das bases de dados AD de cada domínio que integra a floresta. O *global catalogue* é fundamental para permitir operações ao nível da floresta.

Active Directory Domain Controllers

O primeiro *domain controller* a ser criado numa floresta passa a ser o *global catalogue server* contendo essa base de dados, posteriormente, outros *domain controller* da floresta podem também tornar-se *global catalogue servers*.

As diferentes funções *operations master* têm de ser executadas por um único *domain controller (Single-Master Operations)* dentro do nível da estrutura a que se aplicam. Assim:

- Numa floresta: além dos *global catalogue servers*, tem de existir exatamente um DC a exercer a função ***Schema Master*** e um DC a exercer a função ***Domain Naming Master***.

- Num domínio: tem de existir exatamente um DC a exercer a função ***Primary Domain Controller (PDC) Emulator***, um DC a exercer a função ***Infrastructure Master*** e um DC a exercer a função ***Relative ID (RID) Master***.

Quando existem vários DC no mesmo domínio a sincronização do *Active Directory* é garantida (*Multimaster replication*) e é proporcionada tolerância a falhas.

Também é possível definir um DC como sendo *read-only* (**RODC**). Operações como uma pesquisa podem ser realizadas pelo RODC, operações que envolvem alterações não são possíveis. Podem ser instalados em locais remotos de segurança precária e com problemas de ligação aos DC.

Estrutura física do Active Directory - Sites

A infraestrutura lógica de domínios, árvores e florestas não possui qualquer reflexo da realidade física subjacente. Para armazenar informação sobre a localização física é introduzido o conceito de ***Site Object*** no contexto de floresta. Assim sob o ponto de vista físico **a floresta está dividida em sites**.

Isto só acontece porque a floresta é o topo da hierarquia lógica, não há relação direta entre Sites (conceito físico) e domínios, árvores e florestas (conceitos lógicos).

Assim: **um domínio pode estar disperso por vários *Sites***, também: **um *Site* pode fazer parte de vários domínios**.

A relevância de dividir a floresta em sites relaciona-se com a otimização dos mecanismos de replicação entre *domain controllers* dentro da floresta.

A definição dos Sites é realizada com base nas configurações de rede existentes nos DC. Cada ***Site Object*** está associado a um ou mais ***Subnet Objects***, um *subnet object* é uma representação de uma rede de nível 3 (IPv4 ou IPv6). Outros objetos como servidores e workstations estão associados a Sites, com esta informação consegue-se por exemplo determinar para uma workstation qual é o DC do domínio pretendido que está fisicamente mais próximo, eventualmente no mesmo Site.

Active Directory – Organizational Units (OU)

Sob o ponto de vista lógico, um domínio pode ser divido numa estrutura em árvore de ***Organizational Units*** (**OU**).

O objetivo das OU é facilitar a organização e a administração. Uma OU pode conter objetos do domínio tais como utilizadores, grupos, computadores e até outras OU.

As OU devem refletir a estrutura hierárquica ou lógica da organização, cada OU pode por exemplo representar um departamento ou uma secção, é importante que existe alguma relação lógica entre os objetos que se colocam na mesma OU.

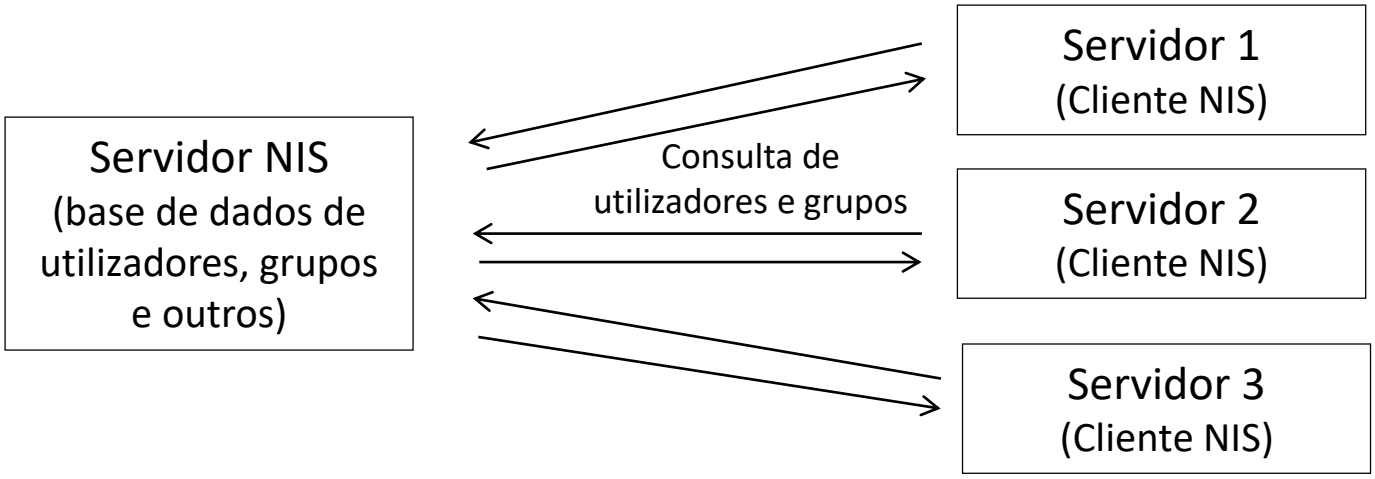
Sob o ponto de vista administrativo destacam-se duas utilizações:

- Uma ***Group Policy*** pode ser aplicada a uma OU tendo efeito sobre todos os objetos contidos na OU.
- Podem ser dadas a utilizadores e grupos permissões sobre uma OU. Desta forma é possível por exemplo criar um administrador de uma OU com permissões para criar utilizadores e grupos nessa OU apenas.

Unix - diversificação dos repositórios do sistema

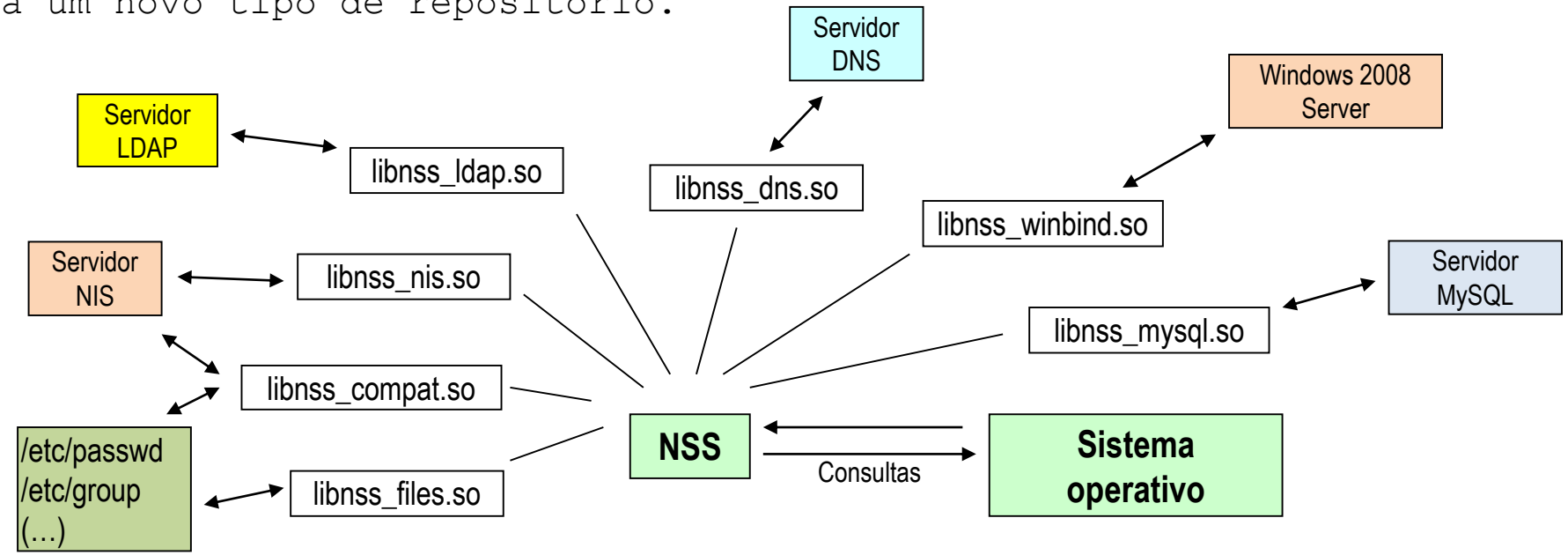
Nos sistemas ***Unix*** desde cedo se sentiu a necessidade de diversificar e centralizar os repositórios de informação de configuração do sistema (entre outros, bases de dados de utilizadores e grupos), essa necessidade levou ao desenvolvimento de sistemas alternativos, um dos mais importantes é o sistema centralizado **NIS** (*Network Information Service*), também conhecido por *Yellow Pages*.

A **centralização** destes repositórios tem enormes vantagens sob o ponto de vista de administração de um conjunto de servidores. Se todos os servidores usarem o mesmo repositório central o administrador consegue gerir toda a informação num único local, os utilizadores e grupos serão os mesmos em qualquer um dos servidores do conjunto.



Name Service Switch (NSS)

O NSS é um sistema modular que permite integrar nos sistemas Linux diversos tipos de repositórios de informação de sistema. Através da adição de um novo módulo (biblioteca dinâmica) acrescenta-se o suporte para um novo tipo de repositório.



Nem todos estes módulos são vocacionados para bases de dados de utilizadores e grupos, por exemplo o *libnss_dns* é normalmente usado para nomes de máquinas.

O módulo *libnss_compat* implementa um modo de compatibilidade com sistemas pré NSS consultando os ficheiros locais e recorrendo adicionalmente a servidores NIS.

Configuração do *Name Service Switch*

A configuração do NSS consiste em definir a ordem pela qual os vários repositórios disponíveis (módulos) vão ser usados. Por motivos de segurança e fiabilidade deve ser dada preferência aos repositórios locais. O ficheiro de texto **/etc/nsswitch.conf** contém a configuração.

Para cada tipo de informação define a sequência de pesquisa. Exemplo:

```
bash-3.00$ cat /etc/nsswitch.conf
#
passwd:     files ldap winbind nis
shadow:     files nis
group:      files ldap winbind nis

hosts:      files nis dns winbind
```

No exemplo as definições de utilizador (***passwd***) serão pesquisadas pela seguinte ordem:

1° Ficheiros locais (***files***)

2° Servidores LDAP (***ldap***)

3° Servidores Windows (***winbind***)

4° Servidores NIS (***nis***)

Será usado o primeiro que for encontrado, terminando a pesquisa

Configuração do *Name Service Switch*

A configuração do NSS propriamente dito é bastante simples, no entanto, muitos dos módulos NSS usados necessitam de configuração específica.

Por exemplo o módulo *libnss_ldap* que permite obter informação em servidores de base de dados através do **LDAP** (*Lightweight Directory Access Protocol*) usa normalmente o ficheiro de configuração **/etc/ldap.conf**. Neste ficheiro, entre outros elementos, vai ter de ser especificado:

- Endereço IP ou nome DNS do servidor LDAP
- DN (*Distinguish Name*) da raiz da base de dados LDAP
- Versão do protocolo LDAP

Ao integrar repositórios remotos de utilizadores e grupos devemos ter consciência de que, o anteriormente referido conjunto de comandos Linux para gestão de utilizadores e grupos, funciona apenas com contas locais armazenadas nos ficheiros **/etc/passwd**, **/etc/shadow** e **/etc/group**.

Em conjunto com o NSS normalmente é também usado o **PAM** (*Pluggable Authentication Module*), os dois não são indissociáveis qualquer um deles pode funcionar sem o outro. Entre outos aspetos, o PAM trata do processo de login dos utilizadores. Quando os utilizadores estão em repositórios remotos o PAM tem de recorrer a eles.

Linux Pluggable Authentication Module (PAM)

O Linux-PAM é usado por todas as distribuições Linux atuais. Trata-se de uma biblioteca modular usada pelas aplicações e serviços que processam as entradas e saídas de utilizadores no sistema.

O PAM trata 4 tipos de tarefas (***management groups***):

- **account** – permite realizar validações antes da autenticação (Exemplos: horário de acesso, endereço de origem) e após a autenticação (Exemplos: acesso permitido para o utilizador, password expirada).
- **auth** – trata-se da autenticação propriamente dita, normalmente através de *username/password*.
- **password** – alteração pelo utilizador da respetiva password.
- **session** – registo e preparação do ambiente. Regista a entrada do utilizador após autenticação bem sucedida, prepara o ambiente de trabalho para o utilizador. Pode ser novamente invocada no *logout*, por exemplo para registar a saída do utilizador.

Existe uma grande variedade de módulos para implementar estas tarefas, muitos deles especificamente desenvolvidos para funcionarem como interface com repositórios remotos. Alguns módulos podem ser usados para vários tipos de tarefas, outros são específicos para determinadas tarefas.

Linux-PAM e autenticação em Linux

A forma tradicional de autenticação de utilizadores em UNIX consiste em comparar o *hash* da *password* que foi fornecida pelo utilizador com o *hash* que está armazenado na conta do utilizador (/etc/passwd ou /etc/shadow).

Com a diversificação de repositórios de contas de utilizador, a forma de realizar a autenticação passou a depender do tipo de repositório. O sistema PAM resolve o problema, tal como o NSS é modular tipicamente para cada módulo NSS existe um módulo PAM para o mesmo tipo de repositório. Por exemplo para LDAP existe o **libnss-ldap** e o **pam-ldap**.

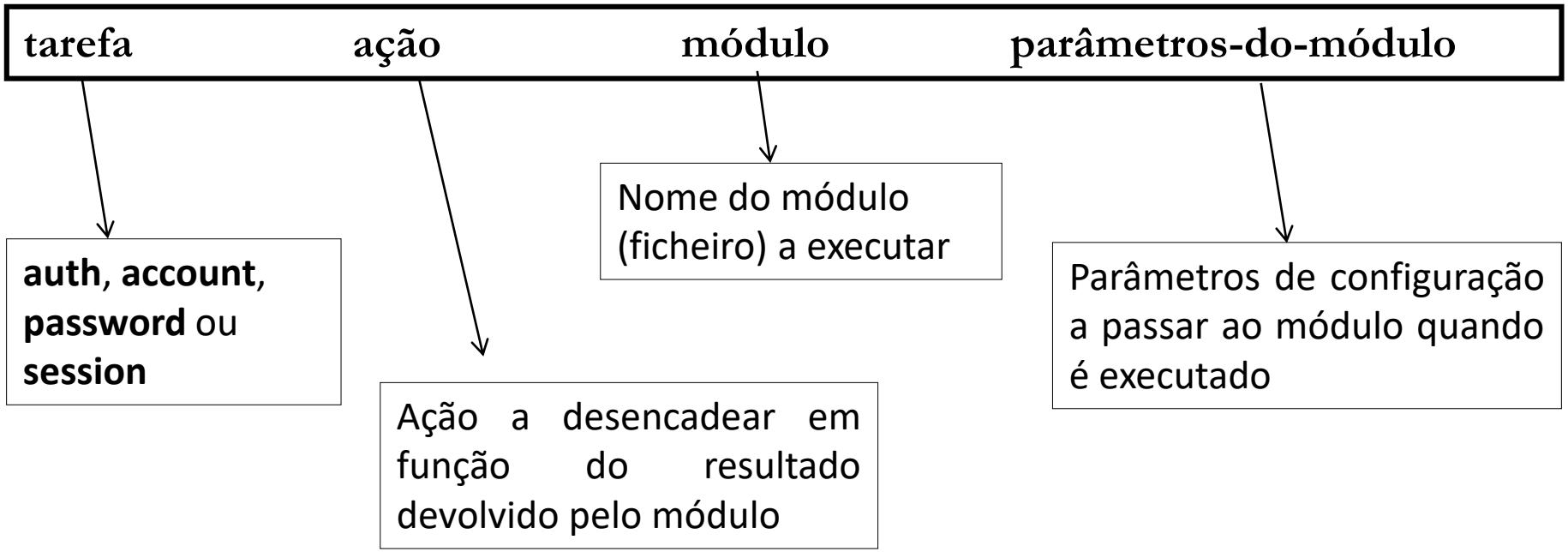
Com o PAM as formas de autenticação podem ser facilmente diversificadas deixando de estar limitadas aos tipos de *hash* suportados nos ficheiros /etc/passwd e /etc/shadow.

O fornecimento direto do par *username/password* ao sistema de autenticação é conhecido por **PAP** (*Password Authentication Protocol*), mas só deve ser usado através de canais de comunicação seguros.

O protocolo **CHAP** (*Challenge Handshake Authentication Protocol*), muitas vezes designado também de autenticação com password protegida, nunca transmite a password legível. O CHAP não pode ser implementado através dos *hash* tradicionais do Unix, mas pode ser implementado com recurso a módulos PAM apropriados.

Linux-PAM - configuração

Configurar o sistema PAM consiste em definir a sequência de módulos a utilizar em cada tipo de tarefa, isso é conseguido com ficheiros de texto, habitualmente residentes na pasta **/etc/pam.d/**, em que cada linha representa a invocação de um módulo no contexto de uma tarefa, na forma:



Os módulos PAM podem devolver vários resultados, o valor zero (**PAM_SUCCESS**) indica sucesso total, outros valores indicam vários tipos de erro. A ação a realizar é definida em função deste valor devolvido pelo módulo.

Linux-PAM – configuração - ação

tarefa	ação	módulo	parâmetros-do-módulo

required – em caso de sucesso continua processamento dos módulos seguintes, em caso de falha também continua a processar os módulos seguintes, mas o resultado final vai ser falha.

requisite - em caso de sucesso continua o processamento dos módulos seguintes, em caso de falha termina de imediato resultando em falha.

sufficient – em caso de sucesso e não existindo falhas *required* anteriores, devolve sucesso e termina a cadeia. Em caso de falha regista uma falha *optional* e continua a processar a cadeia.

optional - em caso de sucesso continua a cadeia, em caso de falha regista uma falha *optional* e continua a processar a cadeia.

Linux-PAM – configuração - ação

A ação a realizar pode ser especificada caso a caso para cada valor possível devolvido pelo módulo:

tarefa	ação	módulo	parâmetros-do-módulo

[value1=ação1 value2=ação2 value3=ação3 …]

Onde **value** representa o resultado devolvido, por exemplo **success** ou **default** e **ação** a ação a realizar nesse caso, as ações podem ser entre outras:

ok – assinala um sucesso e continua a processar os módulos seguintes

done – assinala um sucesso e termina o processamento com sucesso

die – assinala uma falha e termina o processamento com falha

{*NN*} – assinala um sucesso, ignora os *NN* módulos (linhas) seguintes e continua o processamento no módulo (linha) *NN+1*. ({NN} é um número inteiro superior a zero)

Podem referir-se as seguintes equivalências entre formas de especificar a ação:

required	⇔	[success=ok new_authtok_reqd=ok ignore=ignore default=bad]
requisite	⇔	[success=ok new_authtok_reqd=ok ignore=ignore default=die]
sufficient	⇔	[success=done new_authtok_reqd=done default=ignore]
optional	⇔	[success=ok new_authtok_reqd=ok default=ignore]

Linux-PAM – alguns módulos comuns

pam_unix.so	Pode ser usado em qualquer das tarefas, implementa as funcionalidades tradicionais do UNIX pré-PAM.
pam_deny.so	Pode ser usado em qualquer das tarefas, devolve sempre falha. Usado para testes ou para finalizar uma sequência.
pam_env.so	Módulo **auth** e **session** que permite manipular as variáveis de ambiente segundo um ficheiro de configuração, geralmente "/etc/security/pam_env.conf".
pam_mail.so	Módulo **auth** e **account** que faz a verificação de mail no inicio da sessão.
pam_ldap.so	Módulo de interface com repositórios LDAP, suporta as funcionalidades **auth**, **account** e **password**.
pam_issue.so	Módulo **auth** que permite apresentar uma mensagem antes da autenticação do utilizador.
pam_motd.so	Módulo **session** que apresenta uma mensagem após a entrada no sistema (***message of the day***).
pam_nologin.so	Módulo **auth** que devolve sempre sucesso para o administrador e devolve sucesso para os outros utilizadores, a menos que exista o ficheiro **/etc/nologin**.
pam_listfile.so	Pode ser usado em qualquer das cadeias, permite devolver sucesso ou falha em função de uma lista residente num ficheiro, por exemplo com uma lista de utilizadores autorizados.
pam_cracklib.so	Módulo **password** que verifica a solidez da nova *password* que o utilizador pretende usar. Além de a verificar no dicionário do sistema pode também fazer algumas verificações adicionais.
pam_faildelay.so	Módulo **auth** que permite definir o tempo que decorre desde que o utilizador falha a autenticação até que lhe é apresentada a mensagem de erro.

Linux-PAM – utilização pelas aplicações

A configuração do PAM é independente para cada aplicação, cada aplicação é identificada por um **nome de serviço** fornecido à API pela aplicação. Deve existir na pasta **/etc/pam.d/** um ficheiro com o nome de serviço fornecido, será esse o ficheiro de configuração usado.

Exemplo para o serviço SSH que se identifica à API com o nome **sshd**, ficheiro **/etc/pam.d/sshd**:

```
@include common-auth
account  required     pam_nologin.so
account  required     pam_access.so
@include common-account
session    required     pam_loginuid.so
session    optional     pam_keyinit.so force revoke
@include common-session
session    optional     pam_motd.so  motd=/run/motd.dynamic
session    optional     pam_motd.so noupdate
session    optional     pam_mail.so standard noenv
session    required     pam_limits.so
session    required     pam_env.so user_readenv=1 envfile=/etc/default/locale
@include common-password
```

Como se pode observar grande parte das configurações encontra-se em ficheiros incluídos que são partilhados por vários serviços.

Linux-PAM – exemplo /etc/pam.d/common-auth

```
# /etc/pam.d/common-auth - authentication settings common to all services


auth    [success=2 default=ignore]      pam_unix.so nullok_secure
auth    [success=1 default=ignore]      pam_ldap.so use_first_pass
auth    requisite                       pam_deny.so
auth    required                        pam_permit.so
auth    optional                        pam_cap.so
```

Este ficheiro destina-se a ser incluído nas definições **auth** dos vários serviços através da diretiva **@include common-auth.**

Podemos observar que um utilizador será autenticado com sucesso se a autenticação é bem sucedida através dos ficheiros passwd/shadow (**pam_unix.so**) ou a autenticação é bem sucedida através do **bind** (login) no repositório LDAP (**pam_ldap.so**). Se falhar em ambos será atingido o módulo **pam_deny.so** e a autenticação falha.

A informação de configuração do módulo **pam_ldap.so** é a mesma que é usada pelo módulo **libnss_ldap** do NSS, ou seja o ficheiro **/etc/ldap.conf.**