Administração de Sistemas (ASIST)

Aula Teórico Prática 10 - 2018/2019

Linux – serviços de rede através do INETD
Linux – controlo de acesso (*libwrap*)
Windows Server – Network Policy and Access Services
Windows Server - Hyper-V
Windows Server - Emergency Management Services
Windows Server - DCDIAG

Linux – *Standalone Server Daemons*

Para disponibilizar um serviço de rede é necessário manter em execução uma aplicação servidora, permanentemente disponível para receber pedidos de clientes através do protocolo e número de porto estabelecidos para o efeito.

As aplicações que funcionam em background nos sistemas operativos são normalmente conhecida por ***daemons***, muitos deles são aplicações servidoras de rede.

Este tipo de configuração, normalmente designada ***standalone server***, faz todo sentido para serviços de rede com elevado grau de utilização e para os quais é desejável um tempo de resposta o mais curto possível.

É verdade que um ***standalone server***, não consome recursos de CPU porque o processo se encontra normalmente ***idle*** a aguardar um contacto de um cliente, mas consome memória.

Para serviços de rede com baixa taxa de utilização e para os quais o tempo de resposta não é crítico, o ***inetd*** oferece uma alternativa que evita a necessidade de um processo em execução para cada serviço de rede.

Linux - *Internet Service Daemon* (inetd)

O serviço de rede ***inetd*** permite com um único processo em execução, substituir vários ***standalone server daemons*** durante a fase em que estão ***idle*** a aguardar o contacto de um cliente.

Para conseguir esse objetivo o **inetd** cria vários *sockets* UDP e TCP e associa-os os números de porto correspondentes aos serviços que disponibiliza.

Assim que é contactado por um cliente num desses *sockets*, cria um processo filho (*fork()*) e faz-se substituir (*exec()*) pela aplicação servidora correspondente.

É a aplicação servidora que vai efetivamente dialogar com o cliente e não o **inetd**.

Antes de o processo filho ser substituído pelo processo da aplicação servidora efetiva, o descritor correspondente ao *socket* é duplicado para os descritores 0 (*stdin*), 1 (*stdout*) e 2 (*stderr*).

Isso significa que quando a aplicação servidora lê e escreve nestes descritores, na realidade vai estar a ler do *socket* (*stdin*) e a escrever no *socket* (*stdout* e *stderr*).

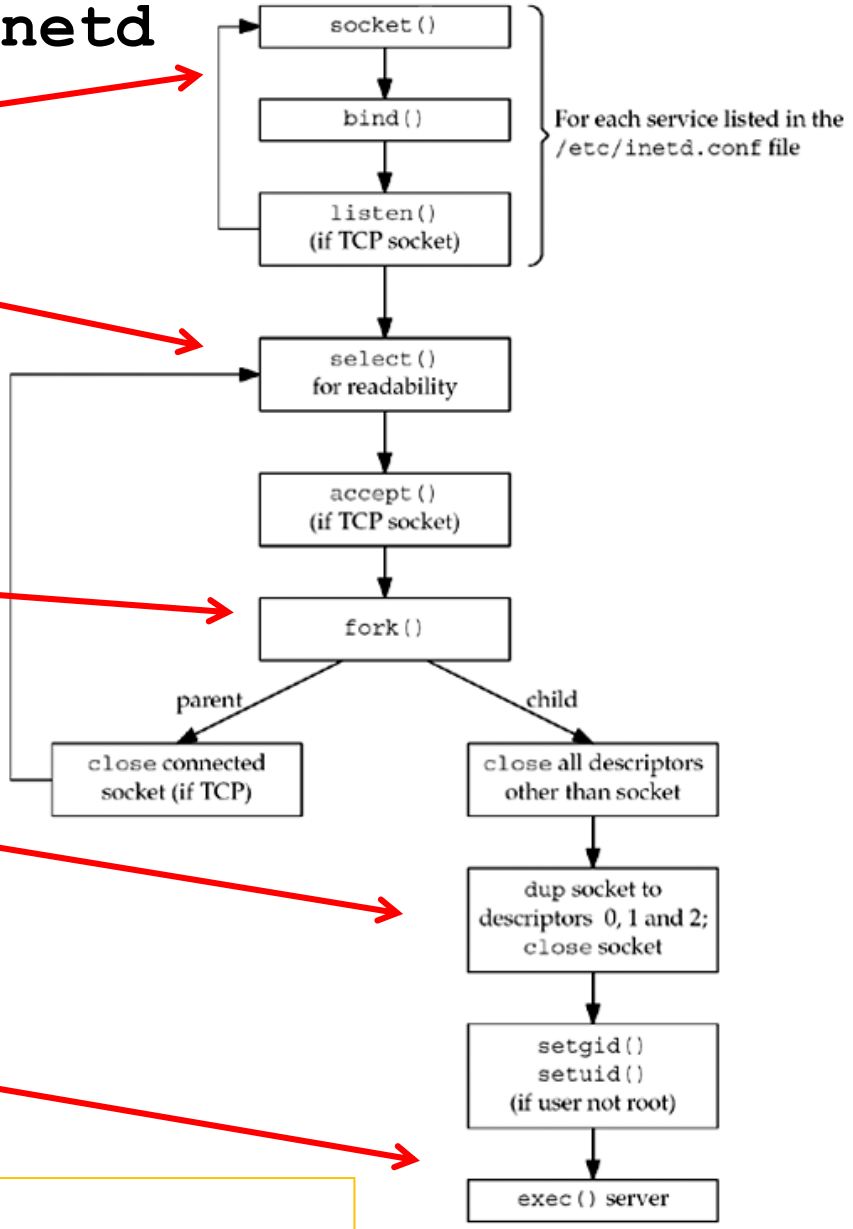
Linux – Funcionamento do inetd

Depois de inicializar os vários *sockets* o INETD recorre à *system-call select* para detetar atividade eu qualquer deles.

Quando um cliente contacta um dos *sockets*, no INETD é criado um processo filho

No processo filho os descritores *stdin*, *stdout* e *stderr* são redirecionados para a ligação de rede (*socket*).

O executável é carregado num ambiente em que os *outputs* são enviado pelo *socket* e os *inputs* são lidos do *socket*.



Source:
http://www.masterraghu.com/subjects/np/introduction/unix_network_programming_v1.3/ch13lev1sec5.html

Linux – inetd – Configuração (/etc/inetd.conf)

O ficheiro **/etc/inetd.conf** define os serviços prestados, é um ficheiro de texto em que cada linha define um serviço, na forma:

service-name	socket-type	protocol	wait/nowait	username	exec-file	args

service-name – é o nome do serviço definido no ficheiro **/etc/services**. Nesse ficheiro ao nome é associado o protocolo (udp ou tcp) e o número de porto.

socket-type – *dgram* para o protocolo UDP ou *stream* para o protocolo TCP

protocol – nome do protocolo de transporte definido no ficheiro **/etc/protocols**, normalmente *udp* ou *tcp*.

wait/nowait – o comportamento ***wait*** obriga o servidor a finalizar o atendimento de um cliente antes de atender o seguinte, é o comportamento típico para serviços UDP, ***nowait*** permite atender vários clientes em simultâneo, é o comportamento típico para serviços TCP.

username – nome do utilizador com que o serviço vai ser executado. Serviços que não necessitam de permissões especiais devem ser executados por um utilizador com poucas permissões, com por exemplo *nobody*.

service-name	socket-type	protocol	wait/nowait	username	exec-file	args

exec-file – caminho desde a raiz até ao ficheiro executável que vai ser usado para dialogar com cliente segundo o protocolo de aplicação.

args – argumentos a fornecer ao executável, incluindo o arg[0].

Exemplo de ficheiro /etc/inetd.conf:

```
ftp      stream  tcp     nowait   root     /usr/bin/ftpd ftpd -1
daytime  stream  tcp     nowait   root     internal
telnet   stream  tcp     nowait   root     /usr/bin/telnetd telnetd
nrpe     stream  tcp     nowait   nagios   /usr/bin/nrpe  nrpe -c /etc/nrpe.cfg --inetd
tftp     dgram   udp     wait     nobody   /usr/bin/tftpd tftpd -s /tftpboot
```

Neste exemplo podemos observar que o segundo serviço é proporcionado autonomamente pelo INETD (*internal*) sem recorrer a nenhum programa externo.

Os nomes dos serviços (à esquerda) devem estar definidos no sistema (/etc/services), cada um corresponde a um número de porto standard para o serviço de rede correspondente.

Linux – Controlo de acesso a serviços de rede

Os sistemas operativos de família Unix incluem uma biblioteca partilhada (*shared library*) que pode ser usada pelas aplicações servidoras para controlo de acesso aos respetivos serviços, trata-se da **libwrap.**

Podemos averiguar se uma aplicação suporta essa biblioteca listando todos os objetos partilhados que o respetivo executável utiliza através do comando **ldd.**

Por exemplo para saber se o serviço SSH (/usr/sbin/sshd) utiliza a biblioteca **libwrap** podemos usar:

```
root@myserver:~# ldd /usr/sbin/sshd
        linux-vdso.so.1 =>  (0x00007ffccedfa000)
        libwrap.so.0 => /lib/x86_64-linux-gnu/libwrap.so.0 (0x00007f22bdfa7000)
        libaudit.so.1 => /lib/x86_64-linux-gnu/libaudit.so.1 (0x00007f22bdd80000)
        libpam.so.0 => /lib/x86_64-linux-gnu/libpam.so.0 (0x00007f22bdb71000)
        libselinux.so.1 => /lib/x86_64-linux-gnu/libselinux.so.1 (0x00007f22bd94f000)
        libsystemd.so.0 => /lib/x86_64-linux-gnu/libsystemd.so.0 (0x00007f22bd8ca000)
        (…)
```

Neste caso conclui-se que o serviço SSH utiliza da **libwrap.**

Linux – libwrap - configuração

A **libwrap** usa dois ficheiros para determinar se o acesso a um serviço deve ser permitido ou não:

- **/etc/hosts.allow**
- **/etc/hosts.deny**

Primeiro é consultado o ficheiro /etc/hosts.allow, em caso de correspondência, o acesso é imediatamente permitido.

Não havendo uma correspondência em /etc/hosts.allow, será consultado o ficheiro /etc/hosts.deny, em caso de correspondência, o acesso é imediatamente negado.

Não havendo nenhuma correspondência em nenhum dos ficheiros o acesso é permitido.

Os ficheiros /etc/hosts.allow e /etc/hosts.deny são constituídos por sequências de linhas na forma geral:

LISTA-DE-DAEMONS : LISTA-DE-CLIENTES

Para cada tentativa de acesso, havendo uma correspondência numa linha o processamento termina e as linhas seguintes são ignoradas.

Os ficheiros **/etc/hosts.allow** e **/etc/hosts.deny** são constituídos por sequências de critérios de correspondência:

LISTA-DE-DAEMONS : LISTA-DE-CLIENTES

Ambas as listas são constituídas por enumerações de valores separadas por espaços ou virgulas. Para que um acesso corresponda a uma linha, terá de ter simultaneamente uma correspondência em ambas as listas da linha.

LISTA-DE-DAEMONS – define uma lista de nomes de processos (argv[0]). Se o serviço de rede é prestado por um processo com um destes nomes, vai corresponder a este critério.

LISTA-DE-CLIENTES – define uma lista de endereços de clientes, pode ser especificada sob a forma de endereços IP, nomes DNS ou padrões. Se o endereço de origem do acesso corresponde a um dos elementos desta lista, vai corresponder a este critério.

Em ambas as listas o valor **ALL** corresponde respetivamente a qualquer *daemon* e qualquer endereço de cliente. Existem várias opções avançadas, por exemplo através do serviço de rede **identd** torna-se possível especificar o nome de utilizador na máquina cliente como critério de acesso a *daemons*.

Linux – suporte de libwrap no inetd

As versões atuais do INETD suportam a libwrap:

```
root@myserver:~# ldd /usr/sbin/inetd
        linux-vdso.so.1 =>  (0x00007ffe831fb000)
        libwrap.so.0 => /lib/x86_64-linux-gnu/libwrap.so.0 (0x00007fcea44cd000)
        libbsd.so.0 => /lib/x86_64-linux-gnu/libbsd.so.0 (0x00007fcea42b8000)
        (…)
```

Se o INETD não suportar a libwrap, ainda assim ela pode ser usada através do **tcpd**, nesse caso o INETD executa o tcpd que por sua vez executa a aplicação servidora.

Exemplos:

```
ftp      stream  tcp     nowait   root     /usr/sbin/tcpd  /usr/bin/ftpd -1
telnet   stream  tcp     nowait   root     /usr/sbin/tcpd  /usr/bin/telnetd
ntalk    dgram   udp     wait     root     /usr/sbin/tcpd  /usr/local/lib/ntalkd
```

Como se pode observar (exemplo ntalk), apesar do seu nome, o tcpd pode ser usado também para serviços de rede UDP.

Windows Server – Network Policy and Access Services

O Windows Server permite realizar vários tipos de validação relativamente aos acessos através da rede.

O role ***Network Policy Server*** **(NPS)** é uma implementação do serviço RADIUS (*Remote Authentication Dial-In User Service*). O RADIUS é um protocolo de autenticação, autorização e registo (AAA – *Authentication, Authorization, Accounting*), não se destina a interagir diretamente com os utilizadores.

Os clientes do servidor RADIUS são serviços de rede que interagem com os utilizadores e transmitem ao servidor as credências de autenticação juntamente com a identificação do serviço, o servidor RADIUS tem como missões:

1º - validar as credenciais do utilizador (***authentication***).

2º - verificar se o utilizador está autorizado a aceder ao serviço identificado (***authorization***).

3º - caso o acesso seja concedido, informa o serviço de que o acesso deve ser concedido e regista-o (***accounting***).

Windows Server – Network Access Protection (NAP)

No Windows Server o NAP é um conjunto de mecanismos que permite realizar vários tipos de validação relativamente aos acessos através da rede.

A máquina cliente deve possui um agente NAP (*NAP-capable computer*) que disponibiliza diretamente ao serviço de rede (*NAP Enforcement Server*) informações sobre o seu estado interno e desta forma o serviço pode determinar se deve ou não autorizar o acesso.

A declaração de estado emitida pelo agente NAP (***SoH – Statement of Health***), inclui tipicamente: estado do firewall, configurações de rede, estado das atualizações de segurança e antivírus, entre outras.

A análise do **SoH** pode ser centralizada, nesse caso o serviço de rede envia-o a um NPS (*Network Policy Server*) configurado para suportar NAP, que desta forma funciona como ***NAP health policy server***.

Vários serviços de rede podem operar como *NAP Enforcement Server*, incluído servidores DHCP e servidores de VPN.

Windows Server – *NAP remediation servers*

Quando uma máquina cliente é declarada ***noncompliant*** pelo NAP, o acesso ao *NAP Enforcement Server*, é negado.

Nas configurações NAP do NPS é possível estabelecer um ***remediation server group*** a ser fornecido aos clientes ***noncompliant***.

O ***remediation server group*** é uma lista de servidores (endereços IP ou nomes DNS) acessíveis aos clientes ***noncompliant*** que estes deverão usar para resolver os seus problema relativos ao NAP e tornarem-se ***compliant***, podem ser entre outros:

- Servidores de assinaturas antivírus.
- Servidores de atualizações (WSUS).
- *Domain Controllers* (problemas relacionados com a falta de autenticação)
- *DNS/DHCP Servers* (se o cliente não está a usar o servidor DNS ou DHCP correto).
- *Troubleshooting servers* (URLs) – páginas com instruções para o administrador do cliente o tornar ***compliant***.

Windows Server - Hyper-V

O Hyper-V é uma plataforma de virtualização que substitui sistemas anteriores equivalentes da Microsoft tais como Microsoft Virtual Server, Microsoft Virtual PC e Windows Virtual PC.

O Hyper-V pode ser ativado nas versões recentes do Windows, nas versões server encontra-se disponível como um role.

A lista de sistemas operativos suportados pelas máquinas virtuais (*guest* OS) inclui inúmeras versões do Windows (server e workstation) e sistemas operativos da família Unix, incluindo o Linux.

Criar uma máquina virtual envolve definir as respetivas características de hardware e o sistema operativo *guest* que se pretende instalar, incluindo: quantidade de memória RAM, discos que vai ter (discos virtuais) e interfaces de rede (associadas a *switches* virtuais definidos no Hyper-V).

Normalmente a instalação do sistema operativo *guest* é realizada a partir de um leitor de CD/DVD virtual associado a um ficheiro com a respetiva imagem (.iso).

Windows Server - Hyper-V – *Switches* virtuais

As interfaces de rede das máquinas virtuais são ligadas a *switches* virtuais definidos no Hyper-V. Um *switch* virtual do Hyper-V pode ser:

- ***External switch*** – está associado a uma interface de rede física do Windows Server onde o Hyper-V está instalado. Permite a comunicação com o exterior através da referida interface física. Isso inclui a comunicação com o próprio servidor, outras máquinas virtuais no mesmo servidor e máquinas virtuais em outros servidores (desde que ligadas a um *external switch*). As máquinas virtuais podem usar diretamente a rede física (*bridge*) ou podem partilhar o acesso à rede definido nas configurações da interface física (NAT). Também é suportada a utilização de VLANs.

- ***Internal switch*** – apenas permite comunicações no interior do servidor, incluindo o próprio servidor e outras máquinas virtuais no mesmo servidor (ligadas ao *switch internal*).

- ***Private switch*** – define uma rede isolada, apenas permite a comunicação entre máquinas virtuais (locais) ligadas ao mesmo *private switch*.

Windows Server - Hyper-V – *Integration Services*

Para garantir uma melhor integração entre o sistema operativo usado na máquina virtual (*guest OS*) e o Hyper-V, existem pacotes de software designados *Integration Services* (**hv_utils**) que podem ser instalados no *guest OS*.

Os *Integration Services* são disponibilizados pelo Hyper-V ao *guest OS* através da opção ***Insert Integration Services Setup Disk*** que irá disponibilizar no leitor de CD/DVD o software adequado para o sistema operativo em causa.

Os *Integration Services* incluem suporte para:

- **Encerramento** (*shutdown*) do *guest OS* através do Hyper-V.
- **Sincronização do relógio** do *guest OS* com o do servidor.
- **Acesso a dados** sobre o estado do *guest OS* pelo Hyper-V.
- ***Heartbeat*** – monitorização do funcionamento do *guest OS* pelo Hyper-V (*watchdog*).
- ***Backup*** (*volume snapshot*) sem necessidade de parar a máquina virtual.
- ***Guest services*** – permitem ao Hyper-V a cópia direta de objetos para os sistemas de ficheiros do *guest OS*.

Windows Server - *Emergency Management Services*

O EMS é uma ferramenta de último recurso que pode ser usada para aceder a um Windows Server que por alguma razão se encontra inacessível ao administrador por não estar a funcionar convenientemente.

Para que esta via de acesso esteja disponível tem de ser ativada nas opções de arranque do sistema operativo (ficheiro BOOT.INI do volume de arranque) este ficheiro deve ser editado indiretamente através do comando **bootcfg** (note-se no entanto que este comando assume que o ficheiro é C:\BOOT.INI o que pode não ser verdade).

Embora o serviço EMS possa ser configurado para ser acessível pela rede, porque nessas situações a rede pode não estar disponível, geralmente opta-se por utilizar uma porta serie, por exemplo:

bootcfg /ems on /port com1 /baud 19200 /id N

Onde **N** representa o número da entrada de arranque no ficheiro C:\BOOT.INI. Com esta configuração, se o servidor ficar inacessível por qualquer outra via pode ser ligado um terminal à porta serie COM1 para interagir através de uma consola.

Windows Server Domain Controller - dcdiag

O **dcdiag** é uma ferramenta de linha de comando que permite realizar testes ao funcionamento de um *Domain Controller* integrado num domínio *Active Directory*.

Sem argumentos realiza um conjunto de testes base, mas pode ser usado para realizar testes adicionais orientados para serviços específicos, por exemplo:

dcdiag /test:DNS

A linha de comando anterior permite realizar vários testes à configuração e funcionamento do serviço DNS (fundamental no contexto de um domínio *Active Directory*).

Esta ferramenta constitui um auxiliar precioso para o administrador permitindo identificar rapidamente problemas de configuração e consistência no seu domínio *Active Directory*.