Criando VServers em Slackware

Silvio Rhatto

22 de agosto de 2007

Resumo

O Linux-VServer é um patch para o kernel que introduz o isolamento de processos do sistema por contextos, permitindo inclusive que se escolha os conjuntos capacidades POSIX permitidas em cada contexto.

Este texto contém o procedimento de construção de vservers utilizando o pacote simplepkg. Existe, no entanto, um procedimento mais artesanal de construção de vservers em slackware na primeira versão deste texto. O embasamento teórico para este tutorial é dado no texo Linux-VServers e segurança por contexto, cuja leitura é recomendada antes que você passe para a parte prática de criar e usar vservers.

1 Possíveis aplicações

- Toolchains automáticos para construção de distros
- Sandbox para ensino de administração de servidores
- Construção de pacotes para não sujar o sistema principal
- Organização de servidores compartilhados por grupos e projetos diferentes
- Replicabilidade de sistemas: basta copiar a pasta do vserver e as configurações
- Isolamento de serviços e aplicações possivelmente inseguras

Isolar os serviços de um servidor talvez seja aplicação mais imediata. Suponha um servidor rodando mySQL, Postfix, Apache com sites em diferentes plataformas (Python, PHP, ...). Havendo uma única brecha em alguma dessas aplicações que permita o ganho de privilégios no sistema, todo ele estará comprometido. Se, pelo contrário, cada aplicação ou conjunto de aplicações estiver isolada uma da outra, as consequências de cada brecha no sistema se restringem muito.

Assim, um servidor contendo um vserver exclusivo para mySQL, outro para sites em PHP, outro para o MTA, outro para sites em Python e assim sucessivamente é uma ótima medida de segurança e organização do sistema.

É possível até construir vservers com uma distro inteiramente compilada contra a uClibc (gentoo embedded, por exemplo), para que o sistema tenha o mínimo de sobrecarga de recursos.

É importante ainda ressaltar que o sistema de vservers está em desenvolvimento e muito provavelmente não oferece *total* (apesar de oferecer bastante) segurança, já que não passou por uma rígida auditoria junto com as demais partes do kernel Linux. Se você realmente quer um nível maior de segurança, use OpenBSD ou FreeBSD jail. Mas para a grande maioria dos sistemas, vservers são uma ótima pedida e facilitam muito o dia-a-dia das tarefas administrativas, contando ainda com a performance do Linux e todas as facilidades das distribuições existentes. Aqui estamos considerando não só a segurança, mas também a performance, portabilidade, organização e overhead do sistema.

2 Procedimento

2.1 Kernel

Baixe, aplique o patch e configure seu kernel. Existem patches para a série 2.4 e 2.6. No config, habilite as opções da seção *Linux Vserver* do seu *menuconfig*. Por questões de estabilidade, desabilite *CONFIG_PREEMPT* e *CONFIG_REGPARM* se você está compilando um kernel da série 2.6. Compile o kernel e instale-o da forma usual.

2.2 Userland

Construa o pacote util-vserver ou baixe-o aqui, juntamente com o pacote vlan.

Se você for construir seu próprio pacote do util-vserver, use a seguinte configuração:

```
./configure --prefix=/usr --sysconfdir=/etc --with-initrddir=/etc/rc.d --localstatedir=/var
```

Depois instale os pacotes:

```
installpkg util-vserver-*.tgz vlan-*.tgz
```

2.3 Script de inicialização

Adicione em algum lugar do seu *rc.M* as linhas

```
if [ -x /etc/rc.d/rc.vservers ]; then
  echo "/usr/lib/util-vserver/vshelper" > /proc/sys/kernel/vshelper
  . /etc/rc.d/rc.vservers start
fi
```

Em seguida, escolha qual script de inicialização você utilizará:

- Vservers que usam o formato antigo de configuração (veja a seguir): */etc/rc.d/vservers-legacy*

- Vservers usando o novo formato de configuração (recomendado): */etc/rc.d/vservers-default*

Uma vez feita escolha, habilite o script correspondente. No nosso caso, utilizaremos o novo formato de configuração:

```
chmod +x /etc/rc.d/vservers-default
ln -s /etc/rc.d/vservers-default /etc/rc.d/rc.vservers
```

Depois de configurar o script de inicialização, reinicie a máquina para que o kernel com suporte a vservers seja carregado.

2.4 Barreira dos /vservers

Agora vamos criar a pasta */vservers* – que armazenará todos os vservers – juntamente com uma barreira contra evasão por *chroot*. Escolha uma partição ou pasta para o */vservers* que suporte atributos extendidos (*ext2*, *ext3*, *reiserfs*, etc). Então dê os comandos:

```
mkdir /vservers
chmod 000 /vservers
chattr +t /vservers
setattr --barrier /vservers
```

Os últimos três comandos são diferentes tipos de barreiras: o primeiro impede que usuários do servidor principal vejam o que há nos vservers – originalmente também foi utilizado como barreira –, enquanto que os dois últimos são sinalizadores para o vserver impedir que alguém de dentro de um vserver faça um chroot para fora – *chattr* para a série 1.x e *setattr* para versões a partir da série 2.x: na dúvida use todos esses sinalizadores, pois você pode mudar a versão do patch e esquecer de atualizar sua barreira.

Para verificar a existência das barreiras,

```
lsattr /
showattr /
```

que deve resultar em algo como

```
----------t- /vservers
```

e

```
---Bui- /vservers
```

2.5 Testando a instalação

Como etapa opcional, caso você queira testar sua instalação, use os scripts de teste disponibilizados pelos desenvolvedores do VServer.

2.6 Criando um vserver

Considerando que o seu simplepkg já esteja instalado e configurado, a instalação de um vserver pode ser feita com o comando

```
ROOT=/vservers mkjail jaula vserver
```

O *mkjail* cuidará do resto. Se você quiser saber o que se passa por detrás das cortinas, leia na primeira versão deste texto o procedimento manual de criação de um vserver. O template *vserver* utiliza o novo formato de configuração dos servidores virtuais. Se você quiser utilizar o formato antigo, crie a jaula com o template *vserver-legacy*.

Se tudo der certo, seu vserver será criado na pasta */vservers/jaula*. Para que ele possa entrar em funcionamento, é preciso antes editar seu arquivo de configuração.

2.7 Configuração da jaula

Existem dois formatos de configuração para um vserver:

- formato de configuração antigo, criado caso você tenha utilizado o template *vserver-legacy* e baseado num único arquivo de configuração
- novo formato e baseado em pastas, criado caso você tenha usado o template *vserver*

Detalhes sobre cada um deles se encontram na seção *Formatos de configuração* do texto Linux VServers e segurança por contexto. Aqui daremos apenas os passos principais para a configuração de rede do seu vserver usando o novo formato de configuração. Isso é feito com a seguinte sequência:

```
cd /etc/vservers/jaula/interfaces
rm -rf * && mkdir jaula && cd $_
echo DEVICE > dev ; echo IP > ip ; echo MASCARA > mask ; echo jaula > name
```

O valor de *DEVICE* é o nome do dispositivo de rede que o vserver utilizará, *eth0* na maioria dos casos. A *MASCARA* é a máscara usada no dispositivo *DEVICE*. O valor de *IP* deve ser:

- Se sua máquina tem um IP real que será dedicado exclusivamente a esse vserver, use-o
- Se você não tem um IP real disponível, *IP* se refere a um endereço numa rede inexistente nas suas atuais conexões, como *10.0.1.1* ou *192.168.1.2* e será atribuido a *DEVICE* através de um alias de rede (detalhes em *Documentation/networking/alias.txt* do código fonte do kernel). Esse IP será usado como uma rede interna e os pacotes que vem ou vão para fora precisam ser roteados via NAT.

Essas configurações já são suficientes para que seu vserver funcione.

3 Iniciando o vserver

Depois de instalar o vserver e arrumar seu arquivo de configuração, é tempo de inicializá-lo:

```
vserver jaula start # liga o servidor virtual
vserver jaula enter # entra no servidor virtual
```

Na primeira vez que você iniciar o vserver, pode ser que ocorram várias mensagens de erro conforme o *init* for executando os scripts de inicialização que tenham comandos relacionados ao hardware do servidor, como montagem de volumes e inicialização da rede. Simplesmente comente essas linhas dos scripts.

Como medida de segurança, mude a senha de root como primeira ação dentro do seu vserver:

```
passwd # mude a senha de root do servidor virtual
exit # pra sair do vserver
```

3.1 Upgrade de vservers

Para o upgrade do seu servidor principal e de todos os vservers que rodam slack, basta usar o *jail-upgrade* do simplepkg.

4 Criando vservers de outras distribuições

A criação de vservers não está limitada à distribuição que a máquina usa. Assim, um servidor slackware pode ser usado para construir jaulas em debian e fedora, usar um *stage* do gentoo ou mesmo a imagem pronta de algum sistema.

Algumas distribuições podem ser instaladas diretamente com o comando *vserver*. A seguir as instruções – retiradas do Debian Grimoire — de como criar um vserver usando debian sarge.

Primeiro, especifique alguns pacotes que não são úteis dentro de um vserver:

```
REMOVE_PACKAGES="sparc-utils,dhcp-client,lilo,makedev,pcmcia-cs,ppp,\
pppconfig,pppoe,pppoeconf,setserial,syslinux,fdutils,libpcap0,iptables,pciutils"
```

Depois, use o comando vserver para criar a jaula usando o *debootstrap*:

```
vserver devel jaula -m debootstrap --hostname satandevel --force -- -d sarge -- \
                                   --exclude=$REMOVE_PACKAGES
```

Isso criará um vserver em debian na pasta */vservers/jaula*: basta arrumar as configurações do vserver e iniciá-lo!

5 Daqui em diante

Neste ponto a maioria das configurações específicas para um sistema tipo Slackware já foram efetuadas. Agora, você precisa configurar seus aplicativos dentro das jaulas. O texto Linux VServers e segurança por contexto contém todos os passos necessários para isso.