Guia de Início Rápido do FreeBSD para Usuários do Linux®

Resumo

Este documento tem como objetivo familiarizar rapidamente os usuários intermediários a avançados do Linux® com os conceitos básicos do FreeBSD.

Índice

1. Introdução . . . 1
2. Shell Padrão . . . 1
3. Pacotes e Ports: Adicionando Softwares no FreeBSD . . . 2
4. Inicialização do Sistema . . . 3
5. Configuração de Rede . . . 4
6. Firewall . . . 5
7. Atualizando o FreeBSD . . . 6
8. procfs: desaparecido, mas não esquecido . . . 6
9. Comandos Comuns . . . 7
10. Conclusão . . . 7

1. Introdução

Este documento destaca algumas das diferenças técnicas entre o FreeBSD e o Linux® para que os usuários intermediários a avançados do Linux® possam se familiarizar rapidamente com os conceitos básicos do FreeBSD.

Este documento pressupõe que o FreeBSD já está instalado. Consulte o capítulo Instalando o FreeBSD no Handbook para obter ajuda com o processo de instalação.

2. Shell Padrão

Usuários de Linux® muitas vezes se surpreendem ao descobrir que o Bash não é a shell padrão no FreeBSD. Na verdade, o Bash não está incluído na instalação padrão. Em vez disso, a shell compatível com o shell Bourne, sh(1), é a shell padrão dos usuários. Por padrão a shell do usuário root é a tcsh(1) no FreeBSD 13 e nas versões anteriores, já no FreeBSD 14 e versões posteriores é a sh(1). A sh(1) é muito semelhante ao Bash, mas com um conjunto de recursos muito menor. Geralmente, os scripts de shell escritos para ash(1) serão executados no Bash, mas o contrário nem sempre é verdadeiro.

No entanto, o Bash e outros shells estão disponíveis para instalação usando a Coleção de Pacotes e Ports do FreeBSD.

Após instalar outro shell, use o chsh(1) para mudar o shell padrão do usuário. É recomendado que o shell padrão do usuário `root` permaneça inalterado, já que shells que não estão incluídas na distribuição base são instalados em /usr/local/bin. Em caso de problema, o sistema de arquivos onde o /usr/local/bin está localizado pode não estar montado. Nesse caso, o usuário `root` não teria acesso ao seu shell padrão, o que impediria o `root` de fazer login e corrigir o problema.

3. Pacotes e Ports: Adicionando Softwares no FreeBSD

O FreeBSD oferece dois métodos para instalar aplicativos: pacotes binários e ports compilados. Cada método tem seus próprios benefícios:

Pacotes Binários

- Instalação mais rápida em comparação à compilação de aplicativos grandes.
- Não requer compreensão de como compilar o software.
- Não é necessário instalar um compilador.

Árvore de Ports

- Capacidade de personalizar as opções de instalação.
- Possibilidade de aplicar patches personalizados.

Se a instalação de um aplicativo não requer personalização, a instalação do pacote é suficiente. Compile o port sempre que um aplicativo exigir personalização das opções padrão. Se necessário, um pacote personalizado pode ser compilado a partir do ports usando o comando `make package`.

Uma lista completa de todos os ports e pacotes disponíveis pode ser encontrada aqui.

3.1. Pacotes

Pacotes são aplicativos pré-compilados, os equivalentes no FreeBSD dos arquivos .deb em sistemas baseados em Debian/Ubuntu e .rpm em sistemas baseados em Red Hat/Fedora. Os pacotes são instalados usando o comando `pkg`. Por exemplo, o seguinte comando instala o Apache 2.4:

```
# pkg install apache24
```

Para mais informações sobre pacotes, consulte a seção 5.4 do Handbook do FreeBSD: Usando o pkgng para Gerenciamento de Pacotes Binários.

3.2. Árvore de Ports

A Coleção de Ports do FreeBSD é um conjunto de Makefiles e patches específicos para instalar

aplicativos a partir do código-fonte no FreeBSD. Ao instalar um port, o sistema buscará o código-fonte, aplicará os patches necessários, compilando o código e instalando o aplicativo e quaisquer dependências necessárias.

A Coleção de Ports, às vezes referida como árvore de ports, pode ser instalada em /usr/ports usando o Git. Instruções detalhadas para a instalação da Coleção de Ports podem ser encontradas na seção 4.5.1 do Handbook do FreeBSD.

Para compilar um port, mude para o diretório do port e inicie o processo de compilação. O seguinte exemplo instala o Apache 2.4 a partir da Coleção de Ports:

```
# cd /usr/ports/www/apache24
# make install clean
```

Um benefício de usar o ports para instalar software é a capacidade de personalizar as opções de instalação. Este exemplo especifica que o módulo mod_ldap também deve ser instalado:

```
# cd /usr/ports/www/apache24
# make WITH_LDAP="YES" install clean
```

Para mais informações, consulte Usando a Coleção de Ports no Handbook do FreeBSD.

4. Inicialização do Sistema

Muitas distribuições Linux® usam o sistema SysV init, enquanto o FreeBSD usa o tradicional estilo BSD de init(8). Sob o estilo BSD de init(8), não há run-levels e o arquivo /etc/inittab não existe. Em vez disso, a inicialização é controlada pelos scripts rc(8). No boot do sistema, o arquivo /etc/rc lê os arquivos /etc/rc.conf e /etc/defaults/rc.conf para determinar quais serviços devem ser iniciados. Os serviços especificados são então iniciados executando os scripts de inicialização de serviço correspondentes localizados em /etc/rc.d/ e /usr/local/etc/rc.d/. Esses scripts são semelhantes aos scripts localizados em /etc/init.d/ nos sistemas Linux®.

Os scripts encontrados em /etc/rc.d/ são relacionados aos aplicativos que fazem parte do sistema "base", como cron(8), sshd(8) e syslog(3). Os scripts em /usr/local/etc/rc.d/ são relacionados aos aplicativos instalados pelo usuário, como o Apache e o Squid.

Como o FreeBSD é desenvolvido como um sistema operacional completo, os aplicativos instalados pelo usuário não são considerados parte do sistema "base". Os aplicativos instalados pelo usuário geralmente são instalados usando Pacotes ou Ports. Para mantê-los separados do sistema base, os aplicativos instalados pelo usuário são instalados em /usr/local/. Portanto, os binários instalados pelo usuário residem em /usr/local/bin/, os arquivos de configuração estão em /usr/local/etc/ e assim por diante.

Os serviços são ativados adicionando uma entrada para o serviço em /etc/rc.conf. As configurações padrão do sistema são encontradas no /etc/defaults/rc.conf e essas configurações padrão são substituídas por configurações definidas no /etc/rc.conf. Consulte rc.conf(5) para obter mais informações sobre as entradas disponíveis. Ao instalar aplicativos adicionais, revise a mensagem de

instalação do aplicativo para determinar como ativar quaisquer serviços associados.

As seguintes entradas em /etc/rc.conf habilitam o sshd(8), habilitam o Apache 2.4 e especificam que o Apache deve ser iniciado com SSL.

```
# enable SSHD
sshd_enable="YES"
# enable Apache with SSL
apache24_enable="YES"
apache24_flags="-DSSL"
```

Depois que um serviço foi habilitado no /etc/rc.conf, ele pode ser iniciado sem reiniciar o sistema:

```
# service sshd start
# service apache24 start
```

Se um serviço não tiver sido habilitado, ele pode ser iniciado a partir da linha de comando usando o comando `onestart`:

```
# service sshd onestart
```

5. Configuração de Rede

Em vez de um identificador genérico *ethX* que o Linux® usa para identificar uma interface de rede, o FreeBSD usa o nome do driver seguido por um número. A saída a seguir do ifconfig(8) mostra duas interfaces de rede Intel® Pro 1000 (em0 e em1):

```
% ifconfig
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=b<RXCSUM,TXCSUM,VLAN_MTU>
        inet 10.10.10.100 netmask 0xffffff00 broadcast 10.10.10.255
        ether 00:50:56:a7:70:b2
        media: Ethernet autoselect (1000baseTX <full-duplex>)
        status: active
em1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=b<RXCSUM,TXCSUM,VLAN_MTU>
        inet 192.168.10.222 netmask 0xffffff00 broadcast 192.168.10.255
        ether 00:50:56:a7:03:2b
        media: Ethernet autoselect (1000baseTX <full-duplex>)
        status: active
```

Um endereço IP pode ser atribuído a uma interface usando o comando ifconfig(8). Para que a configuração de IP permaneça persistente através das reinicializações, ela deve ser incluída no /etc/rc.conf. As seguintes entradas no /etc/rc.conf especificam o nome do host, o endereço IP e o gateway padrão:

```
hostname="server1.example.com"
ifconfig_em0="inet 10.10.10.100 netmask 255.255.255.0"
defaultrouter="10.10.10.1"
```

Use as seguintes entradas para configurar uma interface para utilizar DHCP:

```
hostname="server1.example.com"
ifconfig_em0="DHCP"
```

6. Firewall

O FreeBSD não usa o IPTABLES do Linux® para o seu firewall. Em vez disso, o FreeBSD oferece uma escolha de três firewalls em nível de kernel:

- PF
- IPFILTER
- IPFW

O PF é desenvolvido pelo projeto OpenBSD e portado para o FreeBSD. O PF foi criado como um substituto para o IPFILTER e sua sintaxe é semelhante à do IPFILTER. O PF pode ser combinado com o altq(4) para fornecer recursos de QoS.

Este exemplo de regra do PF permite as conexões de entrada do SSH:

```
pass in on $ext_if inet proto tcp from any to ($ext_if) port 22
```

O IPFILTER é o aplicativo de firewall desenvolvido por Darren Reed. Não é específico do FreeBSD e foi portado para vários sistemas operacionais, incluindo NetBSD, OpenBSD, SunOS, HP/UX e Solaris.

A sintaxe do IPFILTER para permitir o recebimento de conexões no SSH é a seguinte:

```
pass in on $ext_if proto tcp from any to any port = 22
```

O IPFW é o firewall desenvolvido e mantido pelo FreeBSD. Ele pode ser combinado com o dummynet(4) para fornecer recursos de controle de tráfego e simular diferentes tipos de conexões de rede.

A sintaxe do IPFW para permitir a entrada de conexões SSH é a seguinte:

```
ipfw add allow tcp from any to me 22 in via $ext_if
```

7. Atualizando o FreeBSD

Existem dois métodos para atualizar um sistema FreeBSD: a partir do código-fonte ou a partir de atualizações binárias.

Atualizar a partir do código-fonte é o método de atualização mais complexo, mas oferece a maior flexibilidade. O processo envolve sincronizar uma cópia local do código-fonte do FreeBSD com o repositório Git do FreeBSD. Assim que o código-fonte local estiver atualizado, uma nova versão do kernel e do userland pode ser compilada.

As atualizações binárias são semelhantes ao uso do `yum` ou do `apt-get` para atualizar um sistema Linux®. No FreeBSD, o freebsd-update(8) pode ser usado para buscar novas atualizações binárias e instalá-las. Essas atualizações podem ser agendadas usando o cron(8).



Ao usar o cron(8) para agendar atualizações, use o `freebsd-update cron` no crontab(1) para reduzir a possibilidade de que um grande número de máquinas busque atualizações ao mesmo tempo:

```
0 3 * * * root /usr/sbin/freebsd-update cron
```

Para obter mais informações sobre as atualizações a partir do código-fonte e a partir de binários, consulte o capítulo sobre atualizações do Handbook do FreeBSD.

8. procfs: desaparecido, mas não esquecido

Em algumas distribuições Linux®, é possível verificar se o encaminhamento de IP está habilitado verificando o arquivo /proc/sys/net/ipv4/ip_forward. No FreeBSD, o comando sysctl(8) é utilizado para visualizar essa e outras configurações do sistema.

Por exemplo, use o seguinte comando para verificar se o encaminhamento de IP está habilitado em um sistema FreeBSD:

```
% sysctl net.inet.ip.forwarding
net.inet.ip.forwarding: 0
```

Use a opção `-a` para listar todas as configurações do sistema:

```
% sysctl -a | more
```

Se um aplicativo requer o uso do procfs, adicione a seguinte entrada ao arquivo /etc/fstab:

```
proc              /proc          procfs  rw,noauto       0       0
```

Incluir o parâmetro `noauto` faz com que o diretório /proc não seja montado automaticamente durante a inicialização do sistema.

Para montar o sistema de arquivos sem reiniciar o sistema:

```
# mount /proc
```

9. Comandos Comuns

Alguns comandos comuns e equivalentes são os seguintes:

Comando Linux® (Red Hat/Debian)	Equivalente FreeBSD	Propósito
`yum install package` / `apt-get install package`	`pkg install package`	Instala um pacote a partir de um repositório remoto
`rpm -ivh package` / `dpkg -i package`	`pkg add package`	Instala um pacote local
`rpm -qa` / `dpkg -l`	`pkg info`	Lista os pacotes instalados
`lspci`	`pciconf`	Lista os dispositivos PCI
`lsmod`	`kldstat`	Lista os módulos do kernel carregados
`modprobe`	`kldload` / `kldunload`	Carrega / Descarrega módulos do kernel
`strace`	`truss`	Rastrea as chamadas de sistema

10. Conclusão

Este documento forneceu uma visão geral do FreeBSD. Consulte o Handbook do FreeBSD para obter uma cobertura mais detalhada desses tópicos, bem como de muitos outros tópicos não abordados por este documento.