

Instalação de sistemas GNU/Linux em Servidores

Introdução

O objetivo deste documento é mostrar a instalação e configuração de um servidor com o sistema operacional GNU/Linux, podendo ser utilizado como firewall, proxy, VPN, etc. Como o propósito é a segurança, todas as configurações necessárias para tornar este servidor o mais seguro possível serão realizadas.

As distribuições que serão referenciadas neste artigo são: Fedora e SUSE.

Instalação

Geralmente quando realizamos a instalação e configuração de um sistema GNU/Linux que será utilizado como servidor em um ambiente de rede, NÃO é preciso nenhum tipo de interface gráfica, a menos que seja realmente necessário e nenhum serviço ou aplicação além da necessária deve ser instalada.

Obs.: Lembre-se sempre que cada serviço habilitado ou instalado desnecessariamente é um caminho para a intrusão do seu sistema, assim é sempre recomendado realizar a instalação mínima do sistema.

Outro questão importante é sobre o particionamento do disco, onde utilizaremos a seguinte configuração:

Partição	Ponto de Montagem	Tamanho (MB)
/dev/hda1	/boot	100
/dev/hda2	/	1500
/dev/hda3	swap	2x Quantidade Memória RAM
/dev/hda5	/usr	4500
/dev/hda6	-	LVM
/dev/system/var	/var	Depende da finalidade do servidor
/dev/system/home	/home	Depende da finalidade do servidor
/dev/system/tmp	/tmp	< 1000

Obs.: As partições /var, /home e /tmp devem ser criadas com o gerenciador LVM, *Logical Volume Manager*, o que proporciona uma maior flexibilidade para aumentar ou diminuir o tamanho das partições.

Obs.: As partições swap, /var, /home e /tmp podem variar de tamanho dependendo da finalidade de uso do servidor, por exemplo, para um servidor SAMBA, a maior partição deve ser /home, porém para um servidor de Correio Eletrônico que utilize o serviço de Cyrus-IMAP, a maior deve ser a partição /var.

O objetivo da criação destas partições é começar a aplicar segurança a partir do sistema de arquivos, como por exemplo, se o disco for dividido em somente três partições: /boot, / e swap, um invasor

que consiga acesso, mesmo sem ser `root`, pode "derrubar" o servidor simplesmente copiando arquivos para o diretório `/tmp`, onde como todos sabemos tem permissão de escrita para qualquer um, até que o mesmo fique completamente utilizado, fazendo com que o sistema fique sem espaço disponível para nenhuma outra atividade importante.

Outra questão importante é sobre o tamanho da área de `swap`, onde nem sempre o cálculo de multiplicar duas vezes pelo tamanho da memória RAM é o mais correto, por isto é sempre recomendado um estudo antecipado, de acordo com os serviços que estarão sendo disponibilizados. Lembrando ainda que um sistema com freqüente utilização da área de `swap`, significa que temos um problema de falta de memória RAM, proporcionando problemas de performance.

Outra questão importante é o uso de uma senha no gerenciado de boot, no nosso caso o GRUB, onde tal configuração pode ser realizada tanto no momento da instalação, onde eu recomendo, quanto no término.

Para adicionar a senha depois que o sistema já estiver instalado, temos os seguintes passos:

```
# grub-md5-crypt
```

Após digitar a senha, a mesma será impressa na tela, só que criptografada, de acordo com o exemplo abaixo:

```
$L1g#NMd345P.
```

O último passo consiste em copiar a senha e adicionar a seguinte linha no arquivo `/boot/grub/menu.lst`:

```
password --md5 $L1g#NMd345P.
```

Obs.: Mais uma vez, vale lembrar que é recomendado o uso de senhas entre 10 e 12 caracteres, onde podemos forçar este tamanho, alterando o campo `PASS_MIN_LEN` no arquivo `/etc/login.defs` e misturando todos os tipos de caracteres permitidos. A mesma recomendação deve ser utilizada na definição da senha do `root`.

Com o fim da instalação do servidor, vamos armazenar uma lista de todos os pacotes que foram instalados. Para isto usaremos o seguinte comando:

```
# rpm -qa >> pacotes_instalados.txt
```

Em seguida, iremos armazenar uma lista de todos os arquivos com SUID e SGID configurados, com o propósito de realizarmos varreduras futuras, verificando a existência de novos arquivos, que podem ser possíveis backdoors. Para isto, usaremos os comandos abaixo:

```
# find / -type f -perm -4000 >> arquivos_suid.txt  
# find / -type f -perm -2000 >> arquivos_sgid.txt
```

Obs.: É importante que estes arquivos gerados NÃO fiquem somente no servidor, pois um invasor com privilégios de `root` pode facilmente falsificar as informações.

O próximo passo consiste em limitar o acesso ao servidor, onde primeiro vamos editar o arquivo `/etc/inittab`, diminuindo a quantidade de terminais e desabilitando as teclas `CTRL+ALT+DEL`, de acordo com as linhas abaixo:

```
...
# Trap CTRL-ALT-DELETE
#ca::ctrlaltdel:/sbin/shutdown -t3 -r now
...
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
#3:2345:respawn:/sbin/mingetty tty3
#4:2345:respawn:/sbin/mingetty tty4
#5:2345:respawn:/sbin/mingetty tty5
#6:2345:respawn:/sbin/mingetty tty6
...
```

Em seguida, iremos criar um usuário, por exemplo, `sysadmin`, que será utilizado para entrar no sistema.

```
# addgroup sysadmin
# adduser -g sysadmin -c "System Administrator" -m sysadmin
# passwd sysadmin
(Deve ser definida uma senha forte para este usuário)
```

Depois bloquearemos o acesso local para o usuário `root`, comentando todas as linhas do arquivo `/etc/securetty`. Melhorando um pouco, vamos limitar a quantidade de usuários, pertencentes ao grupo `sysadmin`, logados no sistema, através do arquivo `/etc/security/limits.conf`, onde acrescentaremos a seguinte linha:

```
@sysadmin hard maxlogins 2
```

Obs.: A linha acima vai permitir somente dois usuários do grupo `sysadmin` logados no sistema. Lembrando que depois de todos os passos descritos anteriormente, não será mais permitido o acesso local pelo usuário `root`, então é importante testar se realmente o novo usuário consegue acesso ao sistema antes de reiniciar o servidor.

Para finalizar esta etapa, vamos também desabilitar o acesso remoto ao sistema para o usuário `root` através do serviço SSH, através das alterações apresentadas abaixo, que deve ser realizadas no arquivo `/etc/ssh/sshd_config`:

```
# vi /etc/ssh/sshd_config
(Alterar/Adicionar os parâmetros abaixo)
PermitRootLogin no
AllowUsers sysadmin
```

Para ativar as novas definições é preciso reiniciar o serviço SSH:

- Fedora

```
# service sshd restart
```

- SUSE

```
# rcsshd restart
```

O próximo passo consiste em limitar algumas partições, como `/home`, `/tmp` e `/var`, fazendo com que não seja possível, por exemplo, executar qualquer binário ou shell-script copiado ou instalado nestas partições. Para isto, edite o arquivo `/etc/fstab` e acrescente os parâmetros `nosuid`, `noexec` e `nodev` em cada partição, como sugerido abaixo:

```
/dev/mapper/system-home /home ext3 nosuid,noexec,nodev,defaults 1 2
/dev/mapper/system-tmp /tmp ext3 nosuid,noexec,nodev,defaults 1 2
/dev/mapper/system-var /var ext3 nosuid,defaults 1 2
```

Obs.: É importante verificar se a opção não irá afetar o funcionamento do sistema, principalmente em relação a instalação de pacotes, administração dos logs, etc.

Em virtude da opção `noexec` para a partição `/tmp`, um serviço que irá apresentar problemas é o Logrotate, responsável pela administração dos logs do sistema. Para resolver este problema, é necessário adicionar a seguinte linha no arquivo `/etc/cron.daily/logrotate`, conforme abaixo:

```
#!/bin/sh

export TMP=/var/tmp

/usr/sbin/logrotate /etc/logrotate.conf
...
```

Finalizando, vamos atualizar o sistema, com o uso do comando `yum` no caso do Fedora ou com o `YaST` no caso do SUSE, é claro que o mesmo já deve estar acessando a internet para podermos realizar tais procedimentos.

- Fedora

```
# yum update
```

- SUSE

```
# yast online_update
```

Outras Configurações

Para finalizar, é recomendado instalar um IDS, *Intrusion Detection System*, como o Tripwire, que é uma ferramenta desenvolvida para monitoramento das modificações ocorridas no sistema de arquivos, ou o Snort, que funciona tanto pra manter a segurança como para espiar o que se passa pela sua rede. É um prático sniffer que fica "farejando" a rede em busca de pacotes suspeitos e informa nos arquivos de log tudo o que está acontecendo.

Outras configurações importantes dizem respeito aos arquivos `/etc/hosts.allow` e `/etc/hosts.deny`, que são utilizados para controlar o acesso ao servidor, como por exemplo, através de SSH.

Por último, e acredito ser o mais importante, é fundamental a configuração de um firewall, de acordo com os serviços que estiverem sendo disponibilizados pelo servidor.

Referências

Livro: Projeto de Segurança em Software Livre