Práticas de Segurança

Luiz Arthur

“As práticas de segurança apresentadas aqui se referem em sua grande maioria à técnicas de segurança aplicadas em ambientes Linux ou Like-UNIX, mais especificamente sobre o Slackware Linux. Entretanto, todas as práticas apresentadas nesses slides, podem e devem ser aplicadas em outros Sistemas Operacionais, tal como, Windows, FreeBSD, etc.”

Etapas iniciais e fundamentais para prover segurança em computadores

Antes de proteger um sistema computacional em produção, é preciso identificar se ele ainda é o sistema inicialmente proposto que precisava de proteção, ou se, ele já é um **sistema comprometido** e que não possui mais segurança.

É necessário certificar-se que o sistema esteja funcionando corretamente. E depois de proteger o sistema é preciso **encontrar meios** para determinar se as etapas seguidas pelas medidas de segurança estão **mantendo o sistema realmente seguro**.

Portanto, antes de iniciar a proteção em um sistema que já esta em uso, é necessário **verificar se este ainda é seguro**.

Teste o sistema para identificar seu status. Se encontrar evidencia de invasão não-autorizada, presença de malware, presença de um root kit ou evidência de um ataque, use métodos para recuperar o sistema.

Lembre-se, isso é extremamente difícil!!!

A **limpeza e recuperação de sistemas comprometidos**, pode exigir a obtenção e execução de softwares especias, bem como, instruções para remoção de arquivos críticos do sistema. A re-configuração do sistema computacional ou a limpeza do disco rígido pode ser necessário, o que pode deixar o sistema indisponível por um longo período de tempo.

Depois de reinstalar, configurar e colocar na ativa um sistema computacional comprometido, veja se o sistema está operando de forma apropriada e segura.

Antes de tentar **recuperar** um sistema comprometido, é preciso sentar e **contar os custos** e os **resultados finais**.

Leve em consideração o que tem melhor relação custo/beneficio: **reinstalá-lo** ou **recuperá-lo**.

A experiência sugere que o custo real de recuperação de um sistema com a segurança comprometida é muitas vezes mais do que o dobro do custo da estimativa inicial para recuperar este sistema.

Mas não há regras rígidas e rápidas que possam ser usadas para tomar a decisão do que é melhor: recuperar ou reinstalar.

Você terá de pesar o custo, rísco e benefício.

Análise dos sistemas para evidenciar o comprometimento da segurança

Para verificar se o sistema está limpo e não teve a sua segurança comprometida é possível seguir os seguintes passos:

I. Desconexão dos usuários não autorizados;

II. Identificação e fechamento dos processos não autorizados;

III. Verificação de evidência de tentativas de invasão nos arquivos de registros (logs);

IV. Verificação de danos potênciais no sistema de arquivos.

Cada um desses passos devem ser seguidos meticulosamente e investigados. E é justamente estes passos que esses slides irão transcrever.

I. Desconexão dos usuários não autorizados

Os usuários não-autorizados podem estar dentro ou fora da empresa.

A primeira coisa a se fazer neste caso é verificar se ninguém não autorizado tem o **acesso físico** ao sistema, caso isto ocorra proteja fisicamente o acesso ao seu sistema. Computadores não devem ficar em locais públicos da empresa e computadores públicos não devem dar acesso ao sistema da empresa.

O acesso não-autorizado ao sistema deve ser cancelado com extrema urgência, em especial, se o usuário parecer estar ocultando sua identidade.

Em um servidor Linux é possível verificar os usuários ativos através do comando `w`, por exemplo:

```
#w
 14:50:49 up 4 min,  2 users,  load average: 0,22, 0,41, 0,19
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root     tty6     -                14:49    6.00s  0.03s  0.00s top
joao     :0       -                14:47   ?xdm?  15.43s  0.04s startkde
l33t     pts/4    10.0.0.254       15:07    0.00s  0.01s  0.00s lastlog
```

O comando **`w`** produz uma listagem com todos os usuários conectados no sistema. O `w` lista a origem do logon e mostra o processo em execução no momento.

Através do comando `w` é possível verificar algumas anomalias quanto a usuários, por exemplo, um nome de usuário estranho como `l33t` que nunca foi visto antes, pode significar um cracker.

O caso pode ser pior ainda se ele estiver conectado a partir do gateway da rede, e estiver fazendo uso de servidores FTP, Telnet ou do comando `top` para monitorar “quem” e “o que” está no sistema.

O cenário descrito anteriormente significa que a rede pode ter sido invadida e o gateway foi comprometido. É preciso então, tomar uma atitude radical e imediata, pois o intruso pode ter substituído arquivos executáveis do sistema, de modo a ter controle da máquina.

Essa situação exige que a conexão do gateway com o mundo externo (e com a rede local) seja encerrada até que o gateway esteja seguro novamente.

Deve-se também, investigar todos os servidores e clientes na rede interna a fim de identificar qual dano pode ter ocorrido quanto a segurança e as informações da empresa.

Para evitar qualquer risco, é provável que depois de uma possível invasão, reinstalemos todos os servidores internos importantes. Mas, antes de tomar essa decisão devemos aguardar até a questão ter sido investigada a fundo. **A reinstalação imediata do servidor pode apagar informações úteis sobre a invasão e como evitalá fururamente**.

Toda e qualquer evidência de intrusão deve ser relada a polícia, no caso de ter havido atividade criminal.

Como bloquear uma conta de usuário suspeito

Caso tenha-se notado uma atividade suspeita de usuários, tal como, usuários antes desconhecidos, usuários acessando o sistema fora do horário normal de serviço, usuários conectando-se através de máquinas remotas, etc. Todas as contas suspeitas deverão ser bloqueadas. Isto é possível executando o comando `passwd`, com a opção `-l` (do inglês *locked*), por exemplo:

```
# passwd -l nome_do_usuário
```

Contas duplicadas no arquivo `/etc/passwd` que tenham a mesma `UID` (número que identifica o usuário no Linux) também devem ser desativadas. Não é incomum para um intruso configurar uma conta que tenha `UID=0` (uma duplicata da conta `root`), que vise dar acesso ao sistema com privilégio de `root` (o `root` é o usuário administrador Sistema Operacional Linux).

Outra ferramenta útil para a identificação de *logons* de usuário não-autorizados é o comando **`last`**, que relata a atividade do arquivo `/var/log/wtmp`. Todo acesso dos usuários ao sistema deverá ser registrado nesse arquivo. Alguns crackers muitas vezes excluirão este arquivo.

Uma saída do comando `last`:

```
#last
l33t     pts/4        10.0.0.254       Mon Apr 17 15:07   still logged in
luiz     pts/4        localhost        Mon Apr 17 15:06 - 15:06  (00:00)
root     tty6                          Mon Apr 17 14:49   still logged in
```

1 - Atividade

1.1 Fazer uma lista com as principais opções de todos os comando citados no item desconexão de usuários. Qual é a diferença entre se utilizar o `last` e o `w`?

1.2 Descobrir como fazer buscas de usuários com o comando `last` através: do nome do usuário; do terminal de conexão; e através de outros arquivos (tal como, backups) que não o arquivo padrão.

1.3 Para que serve o comando `lastlog` e como este pode ajudar na identificação e monitoramento dos usuários?

1.4 Qual é a diferença do comando `w` e do `who`?

1.5 Existem outras ferramentas/comandos que ajudam no gerenciamento de usuários? Faça uma breve pesquisa na Internet e identifique algumas!

1.6 Quais arquivos estão relacionados aos logs de acesso de usuários no sistema? É possível impedir que estes arquivos de logs sejam apagados?

II. Identificação e fechamento dos processos não-autorizados

Uma vez que o sistema tenha sua segurança comprometida, todos **os aplicativos** que antes eram padrão do sistema computacional, **podem estar alterados** de forma a servir de utilitário **para ataques do invasor**, são os tão famosos cavalos de tróia ou malware.

Para tentar resolver este problema pode-se fazer o **download** dos aplicativos "**originais"** (fonte confiável) e instalá-los novamente, de forma à assegurar que o sistema volte a ser confiável.

Entretanto, a tarefa de se identificar arquivos alterados do sistema (para usos contra a segurança) é uma tarefa extremamente complexa, como a maiorias das tarefas em segurança.

Um bom administrador deve executar aplicativos e comandos para **validar** se os **processos** em execução são ou não **legitmos**.

O comando **ps** do Linux é uma ótima ferramenta para a identificação e compreensão de tarefas que estão sendo executadas pelo sistema.

O comando ps gera uma lista com todos os processos em execução e seus atributos.

Sendo as opções mais freqüentes do comando ps, as que seguem:

a Mostra os processos em execução de todos os usuários. Sem a opção a o ps mostra apenas os processos pertencentes ao usuário que executou o ps.

u Lista processos incluindo nome dos usuários donos dos processos, hora do início das execuções, percentual de CPU, percentual de memória e terminal associado.

x Mostra lista de processos que não têm um terminal associado. Útil para visualizar processos servidores (daemons).

f Mostra os processos em forma de árvore. Muito útil para identificar a relação de processos pai e filho.

Exemplo:

```
#ps aux
Warning: bad ps syntax, perhaps a bogus '-'? See http://procps.sf.net/faq.html
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   668  128 ?        S    14:46   0:00 init [4]
root      4064  0.0  0.2  1520  488 ?        Ss   14:46   0:00 /usr/sbin/inetd
root      4321  0.0  0.5  3356  996 ?        Ss   14:46   0:00 /usr/sbin/sshd
root      4780  0.1  0.5  2068  972 tty6     S+   14:50   0:06 top
root      4863  0.0  0.4  2480  852 pts/3    R+   16:14   0:00 ps -aux
```

O comando **pstree**, também é muito útil para identificar a relação entre processos, pois este mostra toda a árvore de processos desde o `init` até o último processo em execução. É similar ao comando `ps auxf`. Muito útil para o entendimento da hierarquia dos processos no Linux

```
init(1)-+-acpid(4471)
        |-bash(4550)---top(4780)
        |-cardmgr(1272)
        |-crond(4468)
        |-cupsd(4454)
        |-dcopserver(4652)
        |-events/0(3)
        |-gpm(4548)
        |-inetd(4064)
        |-kaccess(4661)
        |-kalarmd(4691)
        |-kded(4656)
        |-kdeinit(4649)-+-artsd(4663)---artsd(4664)
        |               |-kio_file(4683)
        |               |-klauncher(4654)
        |               |-konsole(4687)-+-bash(4703)
        |               |               |-bash(4705)---pstree(4866)
        |               |               `-bash(4701)
        |               |-kwin(4678)
        |               `-soffice(4728)---soffice.bin(4739)---soffice.bin(4740)-+-soffice
```

É comum em um sistema que esteja corrompido que os comandos `top`, `ps` e `pstree` sejam alterados para esconder algum processo malicioso. Assim os **processos** em execução **podem ser checados diretamente no kernel** através da leitura da lista de processos:

```
# cat /proc/*/stat | awk '{print $1,$2}'
```

Identificando especificamente processos e serviços de rede

O comando **`netstat`** é útil para identificar conexões de rede, tabelas de rotas, estatísticas das interfaces, dentre outros.

Assim, o `netstat` é muito útil para a identificação e análise de processos de redes, sendo as suas principais opções: `-c` (habilita o `netstat` para trabalhar de modo continuo, `-i` (lista as interfaces e as estatísticas), `-n` (desabilita a resolução de nomes), `-p` (mostra os processos responsáveis pela execução de serviços da rede), `-r` (exibe a tabela de rotas), `-a` (mostra todas as conexões).

Para descobrir quais processos estão ativos em determinadas portas TCP/IP execute o comando `netstat -ap`.

```
#netstat -ap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 *:time                  *:*                     LISTEN      4064/inetd
tcp        0      0 *:ftp                   *:*                     LISTEN      4064/inetd
udp        0      0 10.0.0.254:netbios-ns   *:*                                 4546/nmbd
udp        0      0 192.168.0.25:netbios-ns *:*                                 4546/nmbd
```

Cada um dos processos ativos em um host deve ser validado. Se um processo não for conhecido fique de sobre-aviso. Se necessário desconecte o sistema até os processos suspeitos serem eliminados ou validados como serviços legítimos.

Um exemplo prático de comprometimento do sistema:

Por alguns dias, o administrador fica perplexo, porque, logo após uma reinicialização do sistema, há um processo chamado **sndme** em execução. Uma varredura no sistema não localiza nenhum arquivo com esse nome.

Quando o comando **netstat -ap** estava em execução, a seguinte linha estava presente na saída:

```
udp    0      0   *:32145    *:*         LISTEN    1118/sndme
```

Isso significa que o processo chamado **sndme** estava aguardando algo acontecer na porta `UDP 32145`. Foi fácil identificar que esse processo pouco usual estava em execução, mas foi um pouco mais difícil descobrir como esse *exploit* conseguia se esconder.

A chave do mistério foi descobrir um arquivo chamado `sendmail.txt`. Uma pesquisa cuidadosa nos arquivos de inicialização isolou um *script shell* chamado durante a inicialização e que fazia o seguinte:

1. Criava um diretório chamado `/var/sndtmp`
2. Copiava o arquivo `sendmail.txt` para este diretório e renomeava para `snd.Z`.
3. Executava o comando `uncrompress snd.Z`

4. Executava o arquivo como *shell script*, executando `sh snd`. Dessa forma, erá extraído um arquivo chamado `snd.c`
5. Compilava o arquivo com o comando `gcc -o sndme snd.c`
6. Vinculava o arquivo `sndme` a `/bin/sndme`
7. Excluia o diretório `/var/sndtmp`
8. Executava `/bin/sndme`
9. Excluia `/bin/sndme`

O processo `sndme` estava aguardando uma mensagem UDP específica que abriria um *root exploit* no sistema. A parte assustadora desse *exploit* é que, de acordo com os *logs* do sistema ele deixou de ser detectado por mais de seis meses.

Não foi encontrado nenhum dano no banco de dados, visto que uma impressão dos dados correspondeu aos registros impressos. Os gerentes dessa empresa não sabem se as informações dos clientes caíram em mãos erradas ou não.

Moral: processos não-autorizados podem ter qualquer origem. Podem ser cavalos-de-tróia plantados por um intruso ou podem ser processos legítimos que não deveriam estar em execução ou que não deveriam estar sendo executados por determinada pessoa.

2. Atividade

2.1 Pesquise através do comando `ps`, todos os processos ativos no sistema, classifique-os pelo uso da CPU e memória.

2.2 Use o comando `pstree` para identificar os processos principais processos do sistema, bem como o processo pai de todos os processos.

2.3 Faça um pequeno tutorial de como utilizar o comando `top` para listar e gerenciar processos. O `top` é um comando parecido com o `ps` porém é mais visual.

2.4 Através do comando `man netstat`, procure mais opções sobre o comando `netstat` que podem ser uteis para identificação e monitoração de processos de rede.

III. Verificação de evidência de tentativas de invasão nos arquivos de log

O arquivos de log (arquivos de log armazenam informações sobre **status do sistema**) do sistema podem ser uma “mina de ouro” de informações sobre o quão seguro está o sistema computacional, dentre outras.

Estes arquivos podem ajudar e muito no gerenciamento de um sistema computacional no Linux. O diretório `/var/log` no Linux que contém os principais logs do sistema, e o principal arquivo de log é o `/var/log/messages`. Entretanto, arquivos de log do sistema como o `/var/log/messages` são muito grandes e podem conter inúmeras informações, o que torna difícil a obtenção rápida e fácil de informações úteis.

Para tentar resolver o problema de obtenção de informações nesses arquivos (que são normalmente texto) é necessário, que á pessoa responsável pela segurança saiba utilizar filtros de texto e procure por termos referentes a segurança, tal como, as palavras-chaves: **`fail`** e **`repeat`**.

Os seguintes comandos podem ser usados para filtrar textos:

```
# grep fail /var/log/messages
# grep repeat /var/log/messages
```

Uma resposta positiva para qualquer das palavras-chaves deve ser investigada.

Um arquivo de log importante no Slackware Linux é o `/var/log/secure`, neste estão armazenados ás tentativas de logon no sistema. O comando `tail` com a opção `-f` ajuda e verificar às ultimas tentativas de logon. Exemplo:

```
# tail -f /var/log/secure
Apr 23 17:57:15 darkstar login[4836]: invalid password for `root'  on `tty6'
Apr 23 17:57:22 darkstar login[4836]: ROOT LOGIN  on `tty6'
Apr 23 22:54:15 darkstar su[4760]: + pts/2 aula-root
Apr 24 14:49:47 darkstar login[4935]: invalid password for `aula'  on `pts/3'
Apr 11 22:07:15 darkstar vsftpd[4997]: connect from 192.168.73.223 (192.168.73.223)
```

Falhas repetidas de login podem ser falhas e **devem ser estudadas**, porém a seqüência de logons podem indicar que é apenas um usuário que se esqueceu de sua senha.

Por isso, faça a sua lição de casa antes de tirar conclusões precipitadas. Entre em contato com os usuários que enfrentaram problema de logon e verifique se eles se lembram do ocorrido. Em caso negativo, você precisará ir mais a fundo até obter a resposta correta. Quando o custo da elucidação do ocorrido realmente parecer fora de controle, leve em conta cada opção e seu custo. Talvez valha mais a pena reinstalar o sistema ou substituí-lo do que encontrar uma resposta totalmente conclusiva sobre por que e como as entradas de log ocorreram. Mas, sempre tente encontrar as respostas.

A configuração e o gerenciamento de arquivos de log serão abordados posteriormente.

3. Atividade

3.1 Procure na Internet quais são os principais arquivos de log do Linux e quais as informações contidas em cada um desses. Localize também os arquivos do Slackware.

3.2 Pesquise quais são as principais palavras chaves a serem estudadas em arquivos de logs do Linux.

3.3 Monte um tutorial de como se utilizar filtros de texto para ajudar na pesquisa de palavras-chaves em arquivos de log.

Complemento da atividade 3.3 - Filtros de texto

O Linux têm diversas ferramentas para trabalhar e transformar arquivos texto puros, sem formatação especial. Estas ferramentas são úteis quando estamos trabalhando com scripts no shell, verificando arquivos de log, etc.

`cat` – O comando `cat` concatena arquivos, imprime seu conteúdo na tela e ainda pode receber texto digitado pelo teclado para um arquivo.

`cut` – Traduzindo ao pé da letra significa colar. Ele lê o conteúdo de um ou mais arquivos e tem como saída uma coluna vertical. Exemplo: `cut -d : -f 1 /etc/passwd`

`head` – Mostra as primeiras 10 linhas do início do texto.

`tail` – visualiza as últimas 10 linhas de um arquivo.

IV. Busca de alterações/danos potenciais no sistema de arquivos

A alteração em sistemas de arquivos normalmente se dá quando o sistema já foi comprometido, por isso é extremamente importante para um administrador saber qual é o status de seu sistema de arquivo.

Por exemplo: um arquivo do sistema pode ser alterado, ter as permissões trocadas, mudar de dono/grupo, ser apagado, ou podem ser acrescentados arquivos que não deveriam existir em um sistema de arquivo (cavalo-de-troia, vírus, etc).

Entretanto o monitoramento de um sistema de arquivos para saber se houve alterações ilícitas e extremamente complexa e requer muito tempo e disposição do administrador da rede.

Infelizmente para busca de alterações em sistemas de arquivos é uma tarefa árdua, pois normalmente não existem ferramentas nativas que fazem este serviço com excelência. Vide RPM (gerenciador de pacotes da RedHat), entretanto não é um padrão em todas as distribuições.

Porém, existem ferramentas que tentam fazer está tarefa, uma ferramenta que merecem destaque é o AIDE .

Verificação de sistemas de arquivos com RPM

A verificação de sistemas de arquivos em sistema Linux que utilizam o RPM (*Red Hat Package Manager*) é relativamente simples. A instalação via RPM contém informações vitais a partir das quais o RPM pode identificar quais arquivos foram alterados desde a instalação. Entretanto, para que isto seja possível tais arquivos tem de serem instalados via RPM caso contrário não terão este controle.

Portanto, é possível comparar a lista dos arquivos no sistema com a lista dos arquivos conhecidos. A partir daí, pode-se obter uma lista dos arquivos que não são do sistema.

Por exemplo, para fazer a verificação do sistema de arquivos execute o comando:

```
# rpm --Va > /tmp/rpmVA.log
```

A saída relativa à execução desse comando consiste em uma linha para cada arquivo que o RPM tenha instalado no sistema.

O formato de cada linha consiste em um campo de status de oito caracteres seguido de um espaço, uma letra c denotando um arquivo, outro espaço, depois o arquivo ou o nome de diretório.

Existe um campo de 8 caracteres que só existira se forem notadas modificações:

`S`	O tamanho do arquivo foi alterado;
`M`	O modo (permissão e tipo de arquivo foi alterado) foi alterado;
`5`	A soma de verificação do MD5 foi alterada;
`D`	As características de um nó de dispositivo foi alterado;
`L`	Um vinculo simbólico foi alterado;
`U`	O proprietário do arquivo/diretório/nó de dispositivo foi alterado;
`G`	O grupo do arquivo/diretório/nó de dispositivo foi alterado;
`T`	O carimbo de hora foi alterado. Se tiver faltando um arquivo a palavra “missing” será impressa no lugar do campo status.

As alterações em binários devem ser consideradas com cautela, se a alteração não puder ser claramente identificada o arquivo deverá ser substituído.

Examinando cada linha do relatório que o RPM gerou, identifique os arquivo de configuração que terão sido alterados durante a configuração do sistema. Os arquivos não devem mudar nunca; eles devem ser colocados em locais conhecidos, como `/bin`, `/sbin`, `/usr/bin`, `/usr/sbin`, `/usr/X11R6`, etc.

Um exemplo de resultado imediatamente acionável seria uma entrada que diz:

.M..... /usr/bin/write

Para identificar pacotes RPM alterados e substituí-lo, o nome do pacote RPM de origem pode ser encontrado por meio da execução de:

```
# rpm –qf nome_do_pacote
```

Através deste comando é possível saber qual pacote contém o arquivo que foi alterado.

Exemplo:

```
# rpm –qf /bin/splash
Bootsplash-1.0-71
```

Isso significa que a reinstalação do pacote `bootsplash` restaurará o conteúdo original do arquivo.

A reinstalação via RPM se da com o comando:

```
# rpm –Uvh --nodeps --force nome_do_pacote_a_ser_atualizado
```

Uma boa prática é, após instalar e configurar o sistema fazer um primeiro relatório e guardá-lo, e periodicamente utilizar o RPM para tirar relatórios dos status dos arquivo e comparar com este primeiro relatório. Desta forma caso haja alterações que não se possa explicar, alguma atitude deve ser tomada.

O **AIDE (*Advanced Instrudion Detection Environment*)** é um programa de detecção de instrusão, mas especificamente é um programa que checa a integridade do sistema de arquivos.

O AIDE **constrói** uma **base de dados**, que é especificada pelo arquivo de configuração do AIDES (`aide.conf`), está base de dados **armazena** vários a**tributos do sistema de arquivos**, incluindo: permissões, número de inodes, usuários, grupos, tamanho dos arquivos, etc. O AIDE também, cria uma verificação criptografica ou hash dos arquivos, o que possibilita a verificação de alteração do sistema de arquivos.

Tipicamente o administrador do sistema irá criar uma base de dados AIDE em um sistema recém instalado (antes de colocalo em produção). Essa primeira base de dados é um gabarito para verificar se o sistema está normal e não foi corrompido.

No arquivo de configuração do AIDE existe a possibilidade de se configurar quais arquivos/diretórios devem ser analisados pelo AIDE (colocados na base de dados), isto é bom para manter uma boa performance do sistema, já que a verificação, principlamente a criptografica consome muito processamento

Assim, a base de dados deve conter informações de arquivos binários, de configuração e bibliotécas. Ou seja, arquivos que não mudam com frequencia e merecem cuidado especial. Conseguentemente, a base de dados AIDE não deve conter arquivos que são alterados frequentemente, tal como, diretórios de usuários, diretórios temporários, spools de emails, etc.

Instalando e configurando o AIDE

Para a instalação do AIDE, basta fazer o download do AIDE no site oficial: `http://www.cs.tut.fi/~rammer/aide.html`. No caso especifico do Slackware Linux, é possível fazer o download de pacotes já compilados no `www.linuxpackages.net`, desta forma é só procurar por AIDE. Também, é necessário resolver duas dependencias no mesmo site procure por `libgcrypt` e `libgpg-error` (procure pelos pacotes mais recentes).

A instalação no Slackware se dá de forma simples com o comando `installpkg`, tal como:

```
#installpkg aide-post0.10cvs20050914rev1.48-i486-1cf.tgz
#installpkg libgcrypt-1.2.2-i486-2cf.tgz
#installpkg libgpg-error-1.0-i486-1arf.tgz
```

Feito a instalação agora basta configurar o arquivo `/etc/aide.conf`.

O arquivo é dividido em três partes:

•**linhas de configuração** - usado para configurar parametros e definir variáveis;
•**linhas de seleção** - indica quais são os arquivos/diretórios que serão adicionados a base de dados AIDE
•**linhas de macro** - define váriaveis do arquivo de configuração.

Um trecho de do arquivo do arquivo `aide.conf`

```
#vi /etc/aide.conf
 # linhas de configuração
 database=file:/var/lib/aide/aide.db
 database_out=file:/var/lib/aide/aide.db.new
 #linhas de macro
 MyRule = p+i+n+u+g+s+b+m+c+md5+sha1
 All=R+a+sha1+rmd160+tiger
 Standard=s+p+u+g+c+md5+sha1
 Min=s+p+u+g+c+sha1
 Minetc=s+p+u+g+sha1
 # linhas de seleção
 /boot Standard
 /lib Standard
 /etc Minetc
 /bin Standard
 /sbin MyRule
 /usr/sbin MyRule
 /root/ Standard
 !/usr/tmp
```

O arquivo `/etc/aide.conf` permite ao administrador criar regras de verificação e indicar quais arquivos serão verificados.

Existem várias opções e regras de verificação disponíveis no AIDE, o administrador poderá criar as suas próprias regras, veja a seguir detalhes sobre elas:

p	Permissão de arquivo: Permissões de arquivos alterados (leitura,escrita, set-userid).
i	inode: Se o inode for alterado, ou danificado e removido.
n	Quantidade de vínculos: Todos os vínculos descritos ao programas é útil para descobrir se comandos como `cp`, `rm`, `ls` foram substituídos por outros, ou se ha agora algum vinculo novo para eles.
u	Propriedade de usuário verifica acesso inadequado a arquivos mudando a propriedade do usuário.
g	Propriedade do Grupo: O mesmo que usuário só que para o grupo.
s	Tamanho do arquivo: Toda a alteração no tamanhos dos arquivos, pode acarretar na mudança do próprio arquivo.
m	Mtime: O horário da última alteração do arquivo, muito pouco usado.
a	Atime: O horário do ultimo acesso ao arquivo, é inseguro pois pode ser burlado.
c	Ctime: A hora de alteração no inode.
S	Alteração no tamanho do arquivo, esperasse que possa aumentar mas nunca diminuir de tamanho.

Algoritmos de verificação:

`md5`	A soma md5, é altamente usado e confiável.
`sha1`	Secure hash algorithm, baseado no sistema de assinatura digital NIST.
`md1600`	O RIPEMD-160, uma função interativa do hash.
`tiger`	Uma função de hash.
`crc32`	A soma CRC32, verifica redundância cíclica. bem rápida, o suporte mhash tem que estar disponível.
`haval`	A soma de verificação haval, também necessita do mhash
`gost`	A soma de verificação gost, também somente disponível com o mhash.

Existem também caracteres de concatenação, tal como:

`E` Grupo Vazio, tudo é ignorado.

`L` Agrupa (`p+i+n+u+g`). Verifica permissões de arquivos, ótimo para logs.

`R` Agrupa (`p+i+n+u+g+s+m+c+md5`), além de verificar as permissões como `L` verifica o tamanho do arquivo, o horário e executa somas md5.

`>` Agrupa (`p+i+n+u+g+S`), mais para alterações no tamanho do arquivo, aceita o seu crescimento, mas não que diminua.

Depois de configurado para iniciar a base de dados do AIDE, execute o comando `aide -i`, para iniciar o banco de dados. Agora toda vez que for necessário um relatório execute o comando `aide -C > resumo.txt`, e o relatório será enviado para o arquivo `resumo.txt`. Talvés antes de executar o comando `aide -C`, seja necessário o comando `cp /var/lib/aide/aide.db.new /var/lib/aide/aide.db`.

Um treixo de uma possível saída do AIDE:

```
Dead symlink detected at /lib/modules/2.4.20-8/build
AIDE found differences between database and filesystem!!
Start timestamp: 2003-07-24 01:19:46
Summary:
Total number of files=97942,added files=1,removed files=4,changed files=15
Added files:
added:/root/.xauthUeIiJl
Removed files:
removed:/etc/aide/.aide.conf.swp
removed:/etc/aide/.aide.conf.swo
Changed files:
changed:/var/gdm
changed:/var/gdm/.gdmfifo
changed:/var/gdm/:0.Xauth
changed:/var/gdm/:0.Xservers
changed:/var/gdm/.fonts.cache-1
Detailed information about changes:
Directory: /var/gdm
 Ctime    : 2003-07-23 22:19:01              , 2003-07-24 00:58:35
File: /var/gdm/.gdmfifo
 Ctime    : 2003-07-23 22:22:55              , 2003-07-24 00:58:35
File: /var/gdm/:0.Xauth
 Ctime    : 2003-07-23 22:18:52              , 2003-07-24 00:57:03
```

```
File: /var/gdm/:0.Xservers
 Ctime    : 2003-07-23 22:22:55              , 2003-07-24 00:58:35
File: /etc/adjtime
 SHA1     : yZTyv3mIu2t9I/ce8sPim7D6du4=     , 4WRwege9sgXkRFweV8o5arUbdf4=

File: /etc/mail/statistics
 Size     : 0                                , 628
 SHA1     : 2jmj7l5rSw0yVb/vlWAYkK/YBwk=     , flGBKHrYP9BrfEjqDF/VEQkpNd0=

Directory: /root/.kde/share/config
 Ctime    : 2003-07-23 22:57:13              , 2003-07-24 01:18:22

File: /root/.bash_history
 Size     : 1545                             , 2209

 Ctime    : 2003-07-23 23:02:11              , 2003-07-24 00:51:27

 MD5      : hYAuZuwJyG958GNXACiWug==         , ChL81IhJuEhlT/WOWSid2A==

 SHA1     : q+esx8Z2eFSs3Wd9JK8LiOpB4Zo=     , f3qgBMg2NcvQsLGtuyUKyQPF9Kk=

File: /root/.fonts.cache-1
 Ctime    : 2003-07-23 22:59:25              , 2003-07-24 00:59:10

File: /root/.ICEauthority
 Size     : 750                              , 561

 Ctime    : 2003-07-23 22:57:25              , 2003-07-24 01:18:34

 MD5      : 5xdtR6HSb2dS8hAxWRcgrw==         , BQqd117SjjyN/U6CdqzloQ==

 SHA1     : uxkTTFs0ERJpC1sXSc/S9Z4fWCg=     , m/lv11HVgSc/GfZkMUiRH8V4qVM=
```

v. Verificação da estabilidade e da disponibilidade do sistema

Um sistema estável é capaz de executar a tarefa solicitada. Um sistema instável usa recursos desnecessariamente e pode causar problemas para outros sistemas.

O sistema também pode ser desativado inesperadamente, apresentar falhas e/ou tornar-se irrecuperável.

Para verificar se há evidencias de instabilidade e corrigi-la, existem dois possíveis passos: a validação de operação de hardware e garantia de estabilidade da energia.

Validação da operação de hardware

As falhas de hardware são normalmente muito fáceis de detectar. Falhas no monitor, no teclado, etc, são muito obvias, pois são aparentes para o usuário. É quase certo que falhas na mídia de armazenamento gerarão logs de erros no arquivo de log do sistema.

Para verificar se existe algum problema na mídia de armazenamento execute o seguinte comando:

```
# grep error /var/log/messages
```

Algumas falhas aparentes de software podem ser causadas por hardware.

Garantia de estabilidade da energia

Fontes de energia instáveis representam grande risco à integridade dos dados.

Os dados do sistema de arquivos podem ser danificados por flutuação de energia, picos e surtos.

E a energia é um fator muitas vezes deixado de lado mas, a maior parte dos danos em hardware ocorre quando a energia é interrompida por pouco tempo e depois volta. O uso de equipamentos de condicionamento de energia, um sistema de energia ininterrupta (UPS – Uninterruptible Power System), é essencial para proteger o equipamento do computador da exposição a tais eventos;

As falhas de hardware mais comuns:

Tipo de falha	**Causa**
Discos rígidos, monitores	Picos e surtos elétricos
Placas-mãe e periféricos	Picos de energia
Portas seriais e interface de rede	Picos de energia e raios

Atividade 4

4.1 Instalar e configurar o AIDE, de forma que este monitore todos os arquivos e diretórios mais importantes do sistema (assim também é preciso pesquisar quais são os arquivos mais importantes dos sistema).

4.2 Pesquise por outras ferramentas que façam a verificação da integridade do sistema de arquivos.

4.3 Verifique a estabilidade e disponibilidade do sistema, bem como pesquise sobre ferramentas que fazem isto de forma automática.

Segurança no acesso à rede

As redes de computadores se tornaram uma ferramenta indispensável na sociedade atual, praticamente todo mundo, direta ou indiretamente está conectado a um tipo de rede. Porém, as redes trouxeram algum problemas de segurança a sistemas de informações, já com a Internet, por exemplo, inúmeras pessoas podem ter acesso a uma máquina na rede.

Então, algumas medidas de segurança devem ser tomadas para prover segurança a máquinas que estejam conectadas a redes de computadores. A seguir iremos ver alguns desses passos.

Desativando serviços desnecessários

O principal motivo para colocar um computador em uma rede é viabilizar a comunicação entre ele e outros computadores. E em geral a preocupação com segurança é uma questão secundária ao implantar um novo servidor. Infelizmente, a configuração correta da segurança de um sistema envolve um acesso muito bem dosado ao sistema.

É necessário conceder apenas o acesso suficiente nem, mais nem memos.

Quando se está protegendo um sistema, a melhor estratégia a ser adotada é limitar a comunicação entre os equipamentos da rede de forma que apenas os serviços realmente necessários sejam fornecidos. A primeira etapa envolvida na limitação da comunicação é permitir que um serviço esteja ativo ou esteja em execução somente quando ele estiver atendendo a um requisito à um cliente que possui permissão a este serviço.

A melhor hora para configurar serviços é logo após a instalação de um sistema computacional. No entanto, no momento em que se está instalando um servidor, pode ser difícil determinar o que é exatamente necessário.

Existe uma sugestão em oito etapas para desativar serviços desnecessários:

1. Retirar a máquina de rede;
2. Identificar os serviços com suporte;
3. Determinar dependências dos serviços;
4. Alterar a configuração do sistema de modo que apenas os serviços necessários estejam ativos;
5. Reinicializar o sistema;
6. Verificar se serviços desnecessários não estão sendo executados;
7. Verificar se os serviços necessários estão sendo executados;
8. Retornar a máquina à rede e verificar a conectividade da rede;

É possível repetir esses passos quantas vezes forem necessários.

1. Retirar a máquina de rede

Se você tiver acabado de instalar o sistema operacional do computador, provavelmente a condição atual representa um risco de segurança desnecessário, já que a grande maioria dos sistemas operacionais são instalados via CD's e tais sistemas depois de seu lançamento apresentam falhas de segurança, que devem ser corrigidas.

A primeira alternativa para se **isolar uma máquina de uma rede** e minimizar a exposição a problemas de segurança, é mais seguro retirar o cabo da rede até os serviços estarem inativos, está é a medida mais drástica.

Se você não puder desconectá-lo, desative temporariamente a interface de rede do servidor, isso é possível executando o seguinte comando:

```
#ifconfig eth0 down
```

Sendo que, o `ifconfig` é responsável por gerenciar os dispositivos de rede, o parâmetro `eth0` indica que estamos desabilitando a primeira placa de rede (mas poderia ser: `eth1`, `eth2`, `ppp0`, `ppp1`, etc) e a opção `down` desliga o dispositivo de rede.

É possível também iniciar o sistema Linux com diversos níveis de execução (*runlevel*), para realizar este serviço, sendo o mais recomendável o modo mono-usuário, também conhecido como *single*.

Veja a seguir os níveis de execução do sistema:

Nível de Execução	**Estado**
`0`	Desligado
`1`	Modo mono-usuário
`2`	Multi-usuário sem rede
`3`	Multi-usuário texto
`4`	Multi-usuário com servidor X (Slackware/BSD)
`5`	Multi-usuário com servidor X (RedHat, SUSE, Debian)
`6`	Reinicialização
`S, single`	Mono-usuário(Slackware)
`M`	Multi-usuário(Slackware)

O *runlevel* pode ser passado durante o *boot* do sistema, sendo normalmente passado no gerenciador de *boot*, após o nome da imagem (kernel) a ser carregado.

Quando o sistema já está ligado e em produção é possível alterar o *runlevel* com o comando `init`, exemplo, `init 1` irá para o nível de execução mono-usuário. É possível verificar o *runlevel* do sistema com o comando `runlevel`.

2. Identificação de serviços necessários

Um fator muito importante em segurança é **determinar quais serviços são realmente necessários**. É importante que se um serviço não seja estritamente necessário, este não deve estar ativo no sistema, ou melhor não deve nem ao menos estar presente no sistema.

Então o administrador da rede deve se perguntar:

- O servidor precisa funcionar como uma unidade de disco compartilhada?
- A máquina é um servidor de Web?
- Precisamos de um servidor de e-mail?
- É necessário o uso de servidores de terminal remoto?

Todos serviço na rede, ou mesmo um programa, ou um comando simples no computador pode ser usado por um hacker para romper a segurança de seu sistema. Isso ocorre porque estes são software e **todo software pode apresentar falhar**, além do que alguns **serviços** podem ser **utilizados incorretamente** causando furos na segurança (por exemplo, o uso errado de um compartilhamento de rede).

Então **a melhor opção é identificar o que é necessário e o que não for deve ser eliminado do computador**. É claro que isto pode causar transtornos futuros, caso o usuário ou o administrador do computador necessite das funcionalidades de um programa que não foi instalado no sistema, mas este é o preço.

A tabela abaixo mostra alguns serviços que podem compor um servidor e podem ou não estar ativos de acordo com suas necessidades:

Serviço	Recomendado	Finalidade
freewnn	Inativo	Mecanismo de conversão japonês
apmd	Inativo	Monitora o status de bateria para laptops
arptables	Inativo	Autoriza um firewall de filtragem de pacotes com arptables
atd	-	Daemon para trabalhos em lote AT
autofs	-	Daemon autofs
canna	Inativo	Suporte a japonês
crond	-	Serviço de trabalho cron
cups	Inativo	Impressora
gpm	Inativo	Mouse no console
hpoj	Inativo	Suporte a HP
ip6tables	-	Firewall ip6tables
iptables	-	Firewall
irqbalance	-	Interrupção em cpu’s multiprocessadores
isdn	Inativo	Drivers ISDN
keytables	-	Configuração de teclado
kudzu	Inativo	Configura novo hardware
mdmonitor	Inativo	Monitoramento RAID
microcode_ctl	-	Aplica-se ao microcodigo da cpu
netfs	Inativo	Monta e desmonta sistemas de arquivos NFS, SMB, NCP
network	-	Configura interfaces de redes
nfslock	Inativo	Daemon para bloqueio NFS
pcmcia	Inativo	BD configurações PCMCIA
portmap	Inativo	Porta DARPA para mapeador de números de portas de programas RPC
random	-	Gerador de números aleatórios
rawdevices	-	Ativa I/O de leitura e gravação
rrhnsd	Inativo	Programa para fazer procuras na rede
sendmail	Inativo	Servidor de e-mail
sgi_fam	-	Daemon para monitoramento de arquivos
sshd	-	Daemon SSH
syslog	-	Daemon para login no sistema
xinetd	-	Daemon para Internet

Além dos serviços que, por padrão, já estão instalados e ativados, existem outros serviços disponíveis. Não os instale e configure amenos que saiba para que são realmente necessários.

Mas **tenha cuidado**: talvez não seja fácil determinar de imediato a real necessidade de um serviço. Pode ser que a função de um dado serviço seja dar suporte a outro serviço ou componente necessário.

Em casos mais extremos não é recomendado a instalação de ferramentas que podem servir para a construção/criação de software, tal como interpretadores, compiladores, etc.

3. Identificação de dependências de serviços

Ás vezes, **determinados serviços requerem que um ou mais daemons sejam executados** (um daemon é um processo executado sem um console associado).

Na tentativa de fazer um servidor funcionar, a maioria das distribuições tende a colocar todos os serviços em execução e operacionais. Desta forma alguns serviços desnecessários após a instalação devem ser desativados.

Um método para resolver este problema é desativar serviço por serviço e verificar se este é necessário, mas isso é muito cansativo. Já que existe um grande número de dependência entre serviços.

Uma alternativa um pouco mais simples para tentar verificar as dependências de pacotes do sistema é utilizar o comando man.

```
# man –k http
CGI::Carp             (3)  - CGI routines for writing to the HTTPD (or other) error log
ab                    (8)  - Apache HTTP server benchmarking tool
apachectl             (8)  - Apache HTTP server control interface
curl_formadd          (3)  - add a section to a multipart/formdata HTTP POST
curl_formfree         (3)  - free a previously build multipart/formdata HTTP POST chain
curl_formparse      (3)  - add a section to a multipart/formdata HTTP POST
http                  (n)  - Client-side implementation of the HTTP/1.0 protocol
httpd                 (8)  - Apache hypertext transfer protocol server
```

O comando mostrá as páginas que contivessem informações que pertencessem ao serviço http. Leia também a seção "*see also*" do comando *man* no fim da página do comando e verifique sobre outras fontes de informações de serviços.

Não é difícil encontrar dependências de serviços nas distribuições Linux compatíveis com o LSB (*Linux Standard Base*) versão 1.3. O LSB requer que os cabeçalhos nos scripts que iniciam e terminam serviços contenham um campo mostrando o serviço do qual cada um depende.

Por exemplo o *ypserver* contém uma linha com **Required-Start: portmap**, que quer dizer que ele precisa do portmap para ser executado. Para maiores informações sobre os cabeçalhos do LSB acesse: www.linuxbase.org.

Outros comandos uteis para verificar o que realmente um processo faz e se ele tem dependências é utilizando os comandos `apropos` e o `whatis`.

O `whatis` procura em uma base de dados descrições dos comandos, as palavras-chave indicadas. Já o `apropos` procura a palavra-chave, numa base de dados contendo breves descrições dos comandos, mostrando todas as descrições onde encontre a referida chave.

4. Desativando serviços desnecessários na incialização do sistema

Após determinar os serviços necessários bem como os desnecessários, o próximo passo é controlá-los para que apenas os serviços necessários sejam iniciados automaticamente no boot e os desnecessários não.

Para isso é preciso alterar os *scripts* de inicialização, alguns serviços estão ligados a *daemons* comuns como o `/etc/rc.d/rc.inetd` ou o `/etc/rc.d/init.d/xinitd`, mas outros possuem *daemons* exclusivos. Nas distribuições Linux os *scripts* de daemons em geral são encontrados dos diretórios `/etc/init.d` ou no diretório `/etc/rc.d`.

Existem várias ferramentas que podem ajudar a alterar *scripts* de inicialização no Linux (ver tabela no slide a seguir).

Distribuição	Ferramenta de configuração de serviços
Debian	rcconf chkconfig
Red Hat	ntsysv serviceconf chkconig tksysv ksysv
Suse	YaST2 chkconfig

O comando `chkconfig` é comum a praticamente todas as distribuições (não no Slackware), é possível ver os serviços ativos com a opção `--list`:

```
# chkconfig --list
```

É possível ver apenas um serviço especifico:

```
# chkconfig --list sendmail
```

Para ativar o `sendmail` nos níveis basta usar o comando:

```
# chkconfig –level 2345 sendmail off
```

Isso irá desabilitar o serviço nos níveis de execução 2345, para ativar basta usar a opção `-- on`.

Ainda é possível desabilitar os serviços editando manualmente os próprios *scripts* de inicialização que normalmente estão associados a diretórios pelos seus números de níveis. Exemplo `/etc/rc.d/rc5d/` faz referencia a serviços iniciados no nível 5.

Quando um sistema é iniciado, ele executa todos os *scripts* que começam com a letra "S" e quanto for desligado o sistema são executados todos os *scripts* que comecem com a letra "K", ainda é respeitado a ordem de execução dos *scripts* através dos números, ou seja, o *script* S99 é executado depois do *script* S98.

Para desativar um serviço basta remover o *link* simbólico:

```
# rm nome_do_arquivo
```

Ex: `# /etc/rc.d/init.d/S01isdn`

A maneira mais fácil de criar um *link* simbólico é entrar no diretório onde o arquivo original esta e digitar:

```
# ln –s origem_destino
```

Ex: `# ln –s /etc/rc.d/init.d/isdn S01isdn`

O Slackware usa o padrão de inicialização BSD-style para iniciar os *scripts System V,* neste nível de inicialização cara *runlevel* é dado por um único arquivo em `/rc/rc.d/rc.*`, cada *runlevel* tem seu próprio diretório e contém seus *scripts* de inicialização.

Basicamente para se ativar ou desativar um serviço no processo de *boot* basta das permissão de execução para este arquivo, por exemplo, se você quer desativar o servidor *ssh* na inicialização basta executar o seguinte comando:

```
#chmod a-x /etc/rc.d/rc.sshd
```

Já para ativar o serviço no processo de *boot* basta atribuir a permissão de execução a este arquivo.

```
#chmod a+x /etc/rc.d/rc.sshd
```

Caso seja necessário que o serviço não seja ativado em um *runlevel* diferente basta editar o arquivo `/etc/rc.d/rc.M`.

5. Reinicialização

Agora que as alterações na configuração de inicialização estão concluídas é necessário reinicializar o sistema para ver se as alterações estão corretas. Execute: `reboot`, `init 6`, ou `shutdown -r now`.

6. Verificação da configuração de serviços desnecessários

Após a reinicialização, verifique se ainda existe algum serviço desnecessário sendo executado, para verificar a configuração de forma manual execute o comando **`service`** (não no Slackware), que retornara os serviços ativos:

```
# service --status –all |grep –v “stopped”
```

Outro método seria usando o *`chkconfig`*`:`

```
# chkconfig | grep –v “off”
```

Para verificar no Slackware os serviços ativos utilize o `ps` concatenado ao `grep`, tal como:

```
#ps -ax | grep sshd
```

O comando anterior verifica se o servidor `ssh` está ativo ou não, caso esteja, verifique novamente se o *script* de inicialização está com permissão de execução, ou se este comando não está sendo chamado em outros *scripts*, como o `/etc/rc.d/rc.local`.

7. Verificação da configuração de serviços necessários.

É possível utilizar três métodos diferentes para determinar se um serviço está sendo executado:

- Verificar a configuração: verifique se tudo está configurado como desejável;
- Testar o serviço: Verifique se os serviços estão ativos, através de varredura de portas;
- Procurar pelo serviço na memória: Esta técnica pode ser usada para determinar se existe algum tipo de programa em execução no momento. Entretanto, essa técnica não informará se o *daemon* da Internet (*indetd*) inicializaria algum outro serviço, porque esses serviços só estarão na memória quando uma atividade de programa cliente inicia-los;

Para verificar se algum serviço está na memória é possível utilizar os comando `ps`, `netstat` ou `lsof`. Por exemplo:

```
# ps aux | grep –v grep |grep sendmail

# netstat –ap | grep –i listen | more

# lsof |grep smtp
```

8. Retorno da máquina à rede

Agora que tudo está certo recoloque o cabo da rede e utilize o comando `ifconfig` para verificar a configuração de rede. Colocando o dispositivo novamente a ativa na rede.

Atividade 5

5.1 – Quais são as formas de se retirar um host de uma rede e torná-lo incomunicável com outras máquinas da rede? Na sua opinião qual é a melhor forma de se retirar um host de uma rede?

5.2 – Quais serviços de rede seriam realmente necessários para que uma máquina Linux fosse cliente de uma rede? Por que?

5.3 – O que são dependências de pacotes no Linux? Quais comando são utilizados para ver dependências de pacotes já instalados? Verifique a dependência do vsftpd.

5.4 – Desative o `sshd` no slackware e ative o `portmap`.

5.5 – O que você faria para eliminar um serviço que continua em execução, mesmo depois de você desativar tal serviço na inicialização do sistema? Como você procuraria o arquivo que está iniciando tal serviço?

Atividade 6

6.1 Instale e configure um sistema operacional Slackware Linux para que este seja um servidor HTTP com um servidor SSH para gerenciamento de tal servidor. Somente estes dois serviços (HTTP e SSH) devem estar presentes nesta instalação, de forma que esta tenha o mínimo de pacotes (software) possíveis, somente o necessários devem estar presentes no sistema.

6.1.1 Identifique todos os serviços presentes no servidor Slackware.

6.1.2 Tente determinar a dependência dos serviços iniciados no servidor Slackware.

6.1.3 Verifique e configure apenas os serviços necessários a serem ativados na inicialização do sistema.

6.1.4 Verifique os serviços que estão na memória.

a