ESTUDO DA APLICAÇÃO DA TÉCNICA DE HARDENING NOS SERVIDORES WEB DO HOSPITAL DE CLÍNICAS DE PORTO ALEGRE*

Belini Fagundes de Mello**

Resumo: Este artigo apresenta um estudo para aplicação da técnica de hardening, blindagem de sistemas operacionais, nos servidores web do Hospital de Clínicas de Porto Alegre - HCPA, com o objetivo de aumentar a segurança e assim reduzir os riscos ao sistema. O trabalho foi realizado por meio de uma pesquisa bibliográfica, procurando obter um roteiro a seguir para identificar e corrigir pontos frágeis na segurança de um sistema GNU/Linux, utilizado como servidor Web. Para tanto, foi aplicado diretamente no servidor web do HCPA as ferramentas de analises e as técnicas de hardening investigadas, apresentando no artigo os resultados obtidos e as soluções realizadas. A pesquisa indicou várias ameaças de segurança possíveis de ocorrer em sistemas recém instalados, ou em sistemas ativos que estejam mal configurados, evidenciando também diversas técnicas para mitigar tais riscos, além da possibilidade de auditar o sistema através de técnicas e ferramentas que evidenciem onde um administrador de sistemas deve atuar para melhorar a segurança em sistemas web GNU/Linux.

Palavras-chave: segurança, servidores, Linux, kernel, lynis, php, mysql, hardening, hospital, hcpa

---

* Artigo apresentado como trabalho de conclusão de curso de Gestão de Segurança da Informação da Universidade do Sul de Santa Catarina, como requisito parcial para obtenção de título de Especialista. Orientador: Prof. Horácio Dutra Mello.

** Acadêmico do curso de Gestão de Segurança da Informação da Universidade do Sul de Santa Catarina. belinifm@gmail.com.

1 INTRODUÇÃO

O Hospital de Clínicas de Porto Alegre – HCPA é uma referência em saúde no Rio Grande do Sul fazendo parte da Faculdade de Medicina da Universidade Federal do Rio Grande do Sul – UFRGS.

Como toda instituição moderna possui um parque de TI, onde precisa se preocupar com a segurança dos seus serviços, processos e ativos. Neste contexto o portal web institucional representa sua identidade virtual, fornecendo informações de interesse do público em geral de pacientes e seus familiares, assim como acadêmicos, fornecedores, gestores do governo e media de comunicação de massa.

Os servidores web que contém os portais da internet e intranet, são baseados no sistema Operacional GNU/Linux Debian e estão ligados a rede do hospital e também na internet. Apesar de haver mecanismos de segurança como firewall, é importante manter o servidor web bem configurado e sem vulnerabilidades para serem exploradas na obtenção de acessos privilegiados tanto no servidor web quanto em sistemas internos da intranet.

Uma maneira de melhorar a segurança é reforçar a atenção nas configurações da pilha de softwares que é instalado no servidor. Identificar quais vulnerabilidades eles possuem e atuar na sua correção.

O hardening é uma técnica de blindagem de sistemas que envolve um processo de mapeamento de ameaças, mitigação dos riscos e execução das atividades corretivas. O principal objetivo é tornar o sistema melhor preparado para enfrentar tentativas de ataques. (BARBOSA, 2012, p. 47).

Devido a relevância da proteção das informações dos servidores web e de impedir que os mesmos sejam usados como caminho para ataques internos tais como acesso ao banco de dados e outros sistemas, é necessário o conhecimento de ferramentas e métodos para aplicar a técnica de hardening nos servidores.

A principal elemento do sistema Linux é o seu kernel e também seu Sistema Operacional, combinados eles são a base para todos aplicativos que

rodam no computador. O sistema Linux e seu kernel são razoavelmente seguros, e possuem uma grande gama de ferramentas e configurações de segurança, além de controle sobre quem, como e quais recursos e aplicações os usuários podem ter acesso.

O maior risco que um sistema Linux pode apresentar é na sua correta configuração, normalmente existe uma quantidade numerosa de elementos para serem ajustados possuindo infinitas configurações, e cada ajuste pode causar sérios problemas de segurança. (MELO, 2014, p. 1).

Neste artigo é apresentado um pouco do conceito e as técnicas de hardening que foram utilizadas para identificar as vulnerabilidades do servidor web do Hospital de Clínicas de Porto Alegre que no momento dos testes utilizava GNU/Linux Debian 8. Também é apresentado as soluções para mitigar as vulnerabilidades encontradas.

2 HARDENING

O processo de fortificação do sistema operacional aplicando técnicas específicas de controles para melhoramento das configurações com o objetivo de tornar mais seguro é conhecido pelo termo em inglês “hardening” em tradução literal “endurecimento”. No contexto da segurança da informação, significa também o processo de proteger um sistema através da redução de suas possíveis vulnerabilidades. (TURNBULL, 2005, p1).

O sistema Linux pode ser um ótimo servidor desde que sejam feitas as configurações e permissões corretas. Segundo (MELO, 2014, p. 2), alguns administradores sem muita experiência em segurança preparam seus servidores com uma instalação básica e depois que suas aplicações já estão funcionando, acabam deixando da maneira que ficou, pois a possibilidade de fazer com que aplicação pare de funcionar realizando um procedimento de segurança é grande.

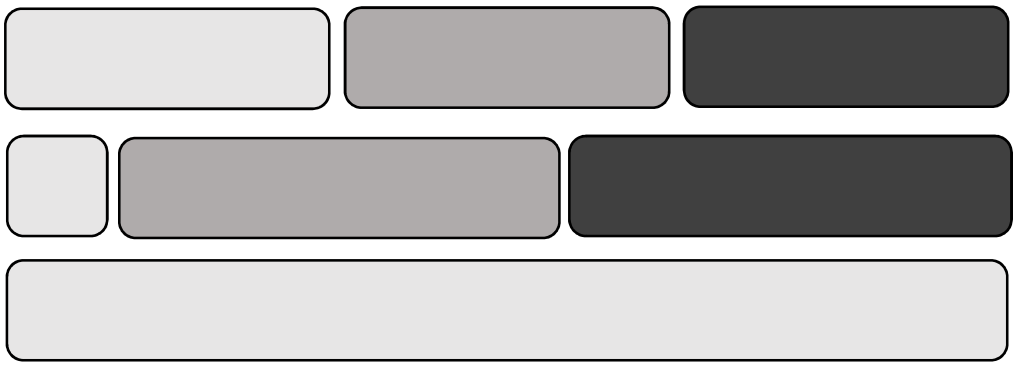
Os Sistemas Operacionais modernos trazem muitos “controles” para melhorar a segurança e gerenciar melhor o uso dos recursos, mas é muito comum que esses controles estejam desabilitados.

Ao planejar a segurança através da técnica de hardening é preciso pensar em três fatores: segurança, risco e flexibilidade, ilustrados na Figura 1: (MELO, 2014, p. 4).

Figura 1 – Representação gráfica três fatores (Segurança, Risco, Flexibilidade)

Onde:

- Implemantação de segurança
- Flexibilidade do usuário
- Risco assumido pela empresa



Fonte: MELO (2014, p. 4).

O analista de segurança envolvido deve dosar bem esses três fatores e levar o sistema a uma alta produtividade, garantindo segurança a um nível aceitável. Ter segurança total diminui completamente os riscos, mas afeta totalmente a experiência do usuário com o sistema. Assim como aumentar a flexibilidade diminui a segurança e aumenta o risco.

Logo, não é possível ter 100% de segurança e o administrador precisa dosar os fatores apresentados analisando cada situação.

No servidor web do Hospital de Clínicas de Porto Alegre - HCPA, onde o autor trabalha como adminstrador web, por exemplo, é utilizado no sistema operacional um usuário com permissões de logon remoto no shell de comandos através de prococolo seguro ssh, assim como por sftp para poder usar aplicativos

da interface de gráfica do sistema remoto e facilitar a gerência e atualização dos arquivos.

No caso do HCPA a política de segurança garante que o usuário tem responsabilidade sobre os dados e uma certa flexibilidade é cedida.

Diante disso, é importante lembrar que ferramentas são importantes, mas não são o fim e sim o meio para a execução dos procedimentos do processo, que, somados a outras ações, como a capacitação das pessoas envolvidas, é fator determinante de sucesso.

A segurança na camada do Sistema Operacional é iniciada no hardening, mas deve estar concordante com as políticas internas de segurança da empresa.

2.1 Analisando o ambiente

Normalmente no processo de hardening de uma nova instalação do SO Linux, diversos fatores são considerados em tempo de instalação, tais como: (COSTA, s.d., passim).

- Desabilitar a inicialização (boot) a partir de dispositivos como drive USB, rede, media ótica (cd/dvd) e até mesmo disquete se for o caso
- Proteger contra mudança nas configurações de hardware (BIOS)
- Certificar-se que o ISO do sistema que vai ser instalado seja de fonte segura
- Planejamento de particionamento dos discos (usar LVM)
- Instalar somente pacotes necessários

O principal foco do artigo é tratar do ambiente existente no servidor web do Hospital de Clínicas de Porto Alegre - HCPA, assim sendo o principal foco será realizar hardening do sistema existente ou já instalado.

Entre os processos para verificação do sistema pós-instalação podemos citar: (MELO, 2014, p6).

- A remoção ou desativação de serviços desnecessários;

- Remoção de contas de usuários-padrão;
- Desinstalação de pacotes desnecessários;
- Definição do processo de atualização;
- Definição de controles para auditoria;
- Definir controles para limites do uso dos recursos pelas aplicações e/ou usuários;
- Instalação de pacotes de ferramentas de segurança e auditoria.

O servidor web do HCPA, possui o seguinte ambiente:

- GNU/Linux Debian 8
- Apache 2.4.10
- PHP 5.6.30
- MySQL 5.5.55

Sendo assim todas técnicas de hardening e os comandos utilizados vão ser aplicadas a um ambiente com configuração semelhante.

2.2. Pacotes instalados

Um bom processo de hardening tem como princípio “menor recurso e menor privilégio” (MELO, 2014, p. 6). Assim deve-se procurar saber se todos pacotes/programas instalados são realmente necessários.

No Quadro 1 é apresentado o comando *dpkg*, utilizado na distribuição GNU/Linux Debian e derivados. Esse comando pesquisa no sistema e lista todos pacotes instalados. O comando *awk* formata a saída, mostrando somente as colunas desejadas em conjunto com o comando *sed* que retira as 5 primeiras linhas.

Quadro 1 – Pesquisando pacotes instalados

```
dpkg -l | awk '{print $2, $3}' | sed '1,5d'


gravando o resultado em um arquivo texto
dpkg -l | awk '{print $2, $3}' | sed '1,5d' > /home/root/pacotes.txt
```

Fonte: MELO (2014, p. 6); COSTA (s.d., p. 14)

Analisar este arquivo pode ser bastante demorado, principalmente por conta da grande quantidade de pacotes instalados no sistema.

Exemplos de programas desnecessários: (MELO, 2014, p. 6); (COSTA, s.d. p. 13).

- lynx: Navegador web via modo texto, possibilita transferência de malware;
- wget: Realiza o download de arquivos de forma recursiva, possibilita transferência de malware;
- netcat(nc): "canivete suiço" que possibilita transferência de malware ou até mesmo backdoors;
- hping: montador de pacotes que possibilita criar backdoors.

Muitos outros aplicativos podem entrar nesta lista, tais como: telnet, cliente ftp, rshd, rlogind, rwhod, ftpd, sendmail, tcpdump, nmap, pois, devido aos recursos que proporcionam, devem ser devidamente avalidados.

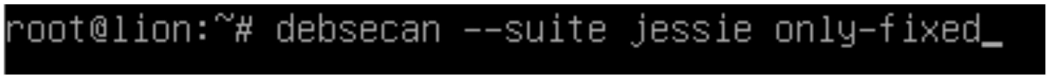
2.3. Avaliação e atualização de segurança

O sistema GNU/Linux Debian possui uma ferramenta que faz avaliação de segurança no ambiente chamada debsecan (*Debian Security Analyzer*). (MELO, 2014, p. 8).

Essa ferramenta verifica a base de pacotes instalada, correlacionando com informações do registro de vulnerabilidades do CVE – *Common Vulnerabilities and Exposures*, que é um padrão mundial utilizado na área de segurança da informação para enumerar vulnerabilidades conhecidas.

O comando debsecan normalmente exibe todos pacotes com notificação de vulnerabilidades, mas nem todos podem ter correção. Para poder encontrar somente os pacotes vulneráveis que já possuem correção pronta pode-se executar o comando da Figura 2, no caso do servidor web do HCPA:

Figura 2 – Executanto debsecan



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Na Figura 3 o print screen das primeiras linhas da saída do comando, onde é possível identificar os pacotes vulneráveis que já possuem correção:

Figura 3 – Saída do debsecan

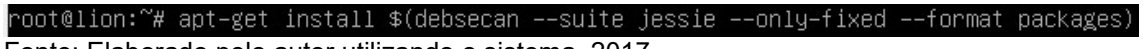
Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Da esquerda pra direita é possível identificar o código CVE da vulnerabilidade, o nome do pacote e um resumo indicando o tipo de vulnerabilidade e a urgência de sua correção.

Esta é uma listagem padrão, mas é possível obter mais detalhes usando outros parâmetros do comando.

O debsecan pode ser utilizado em conjunto com o comando de instalação de pacotes do Debian para atualizar todos pacotes vulneráveis que possuem correção de segurança listados anteriormente, para isso segue na Figura 4 o procedimento para aplicar as correções de vulnerabilidades:

Figura 4 – Aplicando correções de vulnerabilidades



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

2.4. Segurança no sistema de arquivos

O particionamento de discos é um ponto que deve ser levado em consideração na instalação de um sistema Linux, pois além de organizar as informações, podem adicionar mais segurança ao sistema.

> Em um sitema Linux, as boas práticas de instalação recomendam particionar o disco e colocar os principais diretórios em partições separadas. Isso pode proporcionar maior segurança, pois cada partição tem sua tabela separada e pode ter regras de montagem melhor elaboradas. (MELO, 2014, p. 12)

No Linux o comando mount é o responsável por permitir utilizar algumas opções para aumentar a segurança das partições. Crackers podem se aproveitar do diretório "/tmp", por exemplo, onde por padrão qualquer usuário pode escrever, para introduzir backdoors ou qualquer outro programa malicioso para tentar acesso completo ao sistema.

Na Tabela 1 é apresentado um exemplo de como pode ser configurado a tabela de partições e algumas configurações de montagem.

Tabela 1 – Exemplo de particionamento

Ponto de montagem	nosuid	noexec	noatime	nodev
/boot	x	-	-	-
/	-	-	-	-
/home	x	x	-	x
/usr	x	-	-	x
/tmp	x	x	-	x
/var	x	x	-	x
/var/log	x	x	x	x

Fonte: MELO (2014, p. 16); COSTA (s.d., p. 10)

- **Ponto de Montagem**: local onde uma partição é disponibilizada para leitura e gravação de dados;

- **nosuid**: parâmetro usado para inibir que binários com direito especial de suidbit sejam executados em uma partição;
- **noexec**: parâmetro usado para inibir que um binário seja executado em uma partição;
- **noatime**: desativa o registro de tempo/data de acesso dos arquivos (denominado atime);.
- **nodev**: tira o suporte a arquivos de dispositivos;

Segue um exemplo, no Quadro 2 de como essa tabela ficaria no arquivo de configurações de partições /etc/fstab.

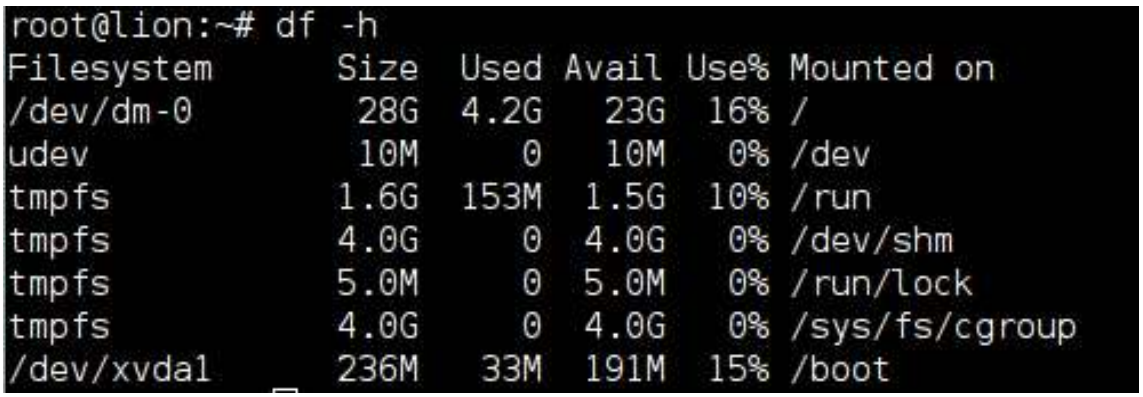
Quadro 2 – Listagem do fstab

```
# /etc/fstab: static file system information.
#<file system> <mount point> <type> <options>                    <dump><pass>
/dev/hda1      /boot      ext3   defaults,nosuid                        0 2
/dev/hda3      /          ext3   defaults                               0 1
/dev/hda4      /home      ext3   defaults,nosuid,noexec,nodev           0 2
/dev/hda5      /usr       ext3   defaults,nosuid,nodev                  0 2
/dev/hda6      /tmp       ext3   defaults,nosuid,noexec,nodev           0 2
/dev/hda7      /var       ext3   defaults,nosuid,noexec,nodev           0 2
/dev/hda8      /var/log   ext3   defaults,nosuid,noexec,noatime,nodev   0 2
```

Fonte: MELO (2014, p. 16); COSTA (s.d., p. 10)

Um comando Linux utilizado para visualizar as partições montadas é o *df -h*, na Figura 5 é mostrado a execução do mesmo no servidor www do Hospital de Clínicas de Porto Alegre - HCPA.

Figura 5 – Montagem do sistema de arquivos no servidor web do HCPA



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Como pode ser visto na primeira linha, no caso deste servidor web, toda instalação está concentrada em uma partição, no diretório “/” (barra), apenas o /boot, está em partição separada.

As outras partições com sistema de arquivos tmpfs são usados como RAMDISK, para fins de performance.

Veja no Quadro 3 o arquivo /etc/fstab, onde consta as configurações de partições:

Quadro 3 – fstab do servidor web do HCPA

```
# /etc/fstab: static file system information.
# <file system>          <mount point> <type><options>       <dump><pass>
/dev/mapper/lion--vg-root       /       ext4   errors=remount-ro  0      1
/dev/xvda1                      /boot   ext2   defaults           0      2
/dev/mapper/lion--vg-swap_1     none    swap   sw                 0      0
```

Fonte: Elaborado pelo autor baseado no sistema, 2017.

Neste caso específico não possui muitas opções de configuração de segurança possíveis a nível de ponto de montagem, a melhor opção seria reparticionar o disco para poder usar os comandos de montagem adequados para segurança.

Adicionalmente poderia-se incluir a opção nosuid na montagem do / (barra), mas neste caso eu impediria que usuários usados por administradores

de rede tivessem opção de executar diversos comandos necessários para seu uso diário, visto que toda montagem perderia permissão de execução de binários para outros usuários que não fosse o root.

2.5 Segurança por Suid bit

A permissão Suid bit possibilita que um determinado binário possa ser executados por outros usuários pertencentes ao mesmo grupo do usuário dono (owner) do arquivo. Assim se o usuário de um determinado binário for o root, e esse binário tiver ativado o direito de suid bit, ele poderá ser executado por outro usuário comum do sistema com permissões semelhantes ao root.

> Mas qual o problema de se ter um Suid bit ativado. O problema pode ser muito grande se um cracker souber aproveitá-lo, explorando vulnerabilidades conhecidos para conseguir uma shell do root. Podemos pensar no exemplo clássico das shells com permissão de suid bit que quando executada por um usuário, pode liberar uma shell de root. (BARBOSA, 2012, p. 55)

Muitos binários do sistema possuem a permissão de Suid bit ativado, devido ao fato de que alguns comandos podem ser necessários para o uso por usuários comum. Alguns exemplos são o *su*, o *ping*, o *ifconfig* e o *passwd*.

De qualquer forma uma boa prática é manter o acesso à informação e às funções do sistema restritas de acordo com as políticas de segurança e controle de acesso. “Dessa forma, a remoção da permissão de Suid bit dos diversos binários que a possuem deve ser mais um procedimento de hardening.” (MELO, 2014, p. 10)

Inicialmente, para remoção do Suid bit dos binários, precisamos identificar quais binários tem esta permissão, assim precisamos gerar uma lista para melhor análise.

Executando o comando da Figura 6 no servidor web, será gerado uma lista dos binários que possuem a permissão de suid bit:

Figura 6 – Listar binários com permissão Suid bit

Fonte: Elaborado pelo autor utilizando o sistema, 2017.

O número 4000 representa a permissão de Suid bit, sendo que os três zeros são as permissões padrões do sistema (0 para usuário, 0 para grupo e 0 para outros), e o 4 representa a permissão a permissão Suid bit.

O comando da fifura 6 gera a seguinte listagem do Quadro 4:

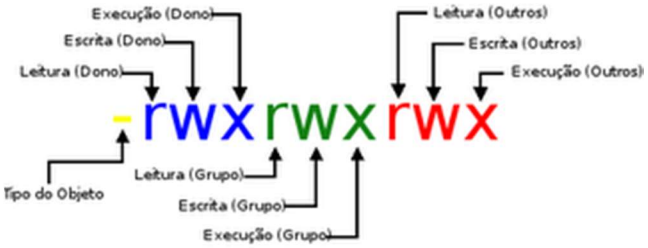
Quadro 4 – Arquivos com Suid bit ativado

/usr/lib/openssh/ssh-keysign	/usr/bin/sudo
/usr/lib/eject/dmcrypt-get-device	/usr/bin/chfn
/usr/lib/dbus-1.0/dbus-daemon-launch-helper	/usr/bin/newgrp
/usr/bin/gpasswd	/usr/bin/procmail
/usr/bin/at	/var/www/html
/usr/bin/ksu	/sbin/mount.nfs
/usr/bin/passwd	/bin/umount
/usr/bin/chsh	/bin/mount

Fonte: Elaborado pelo autor baseado no sistema, 2017.

Lembrando as permissões básicas, na Figura 7 e seu significado no Quadro 5: (SILVA, 2006, passim).

Figura 7 – pemissões de diretótrios e arquivos Linux



Fonte: SILVA (2006, passim)

Sendo que os três primeiros rwx pertencem ao dono do arquivo, os outros três rwx pertencem ao grupo e por fim os últimos três rwx pertencem há outros usuários que não fazem parte do grupo, ou seja, outros.

Quadro 5 – Refrência das permissões básicas Linux

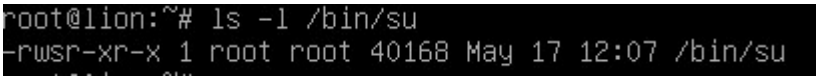
O Tipo do Objeto significa:	Já os outros caracteres significam:
d - diretório; b - arquivo de bloco;	r - permissão de leitura (read); w - permissão de gravação (write);

c - arquivo especial de caractere; p - canal; s - socket; - - arquivo normal.	x - permissão de execução (execution); - - permissão desabilitada.

Fonte: SILVA (2006, passim).

Na Figura 8 é feito a analise detalhada de um dos binários listado no Quadro 4.

Figura 8 – Detalhes do binário

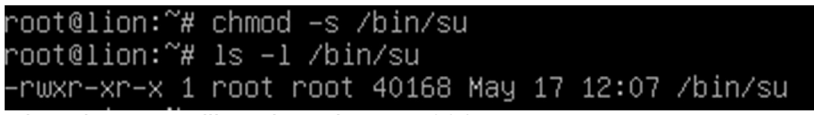


Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Neste caso o grupo de permissões do dono do arquivo mostra um “s” no lugar do “x”, isto indica que a permissão de Suid bit está ativada. A permissão suid bit pode aparecer somente no campo dono (owner).

Na Figura 9 é executado o comando para remover a permissão de Suid bit, *chmod –s*, e após listando para ver o resultado:

Figura 9 – Removendo Suid bit



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Agora aparece o “x” no lugar do “s” mostrando que o comando */sbin/su*, não poderá mais ser executado através do shell de um usuário comum.

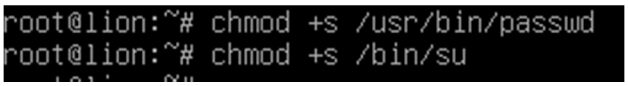
Para retirar a permissão de Suid bit de todos binários do sistema é preciso executar o comando da Figura 10.

Figura 10 – Removendo Suid bit de todos binários

Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Logo após remover o suid bit de todo o sistema, basta definir a permissão somente para os binários que julgarmos realmente necessário, exemplo na Figura 11.

Figura 11 – Incluindo Suid bit em binários



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Pode ser que para um determinado servidor só o *passwd* e o *su* não sejam suficientes. Provavelmente outros binários precisem estar com o Suid bit ativado. Cabe ao administrador analisar o que será necessário para cada tipo de serviço. Uma solução alternativa ao Suid bit é a adoção do comando *sudo*.

2.5 Segurança com o comando sudo

O aplicativo *sudo* executa comandos como outro usuário, desde que a configuração do mesmo seja efetuada corretamente.

Este módulo não torna o código mais ou menos seguro, ele simplesmente usa um mecanismo diferente para ser executado como um outro usuário. Dessa forma o uso do comando sudo acaba sendo uma alternativa interessante ao uso da permissão especial de Suid bit. (TURNBULL, 2005, passim).

Sudo seria algo como – Super User Do, ou em tradução literal, super usuário faça, a vantagem é poder conceder a usuários restritos o poder de executar comandos como se fosse o super usuário ou root do sistema.

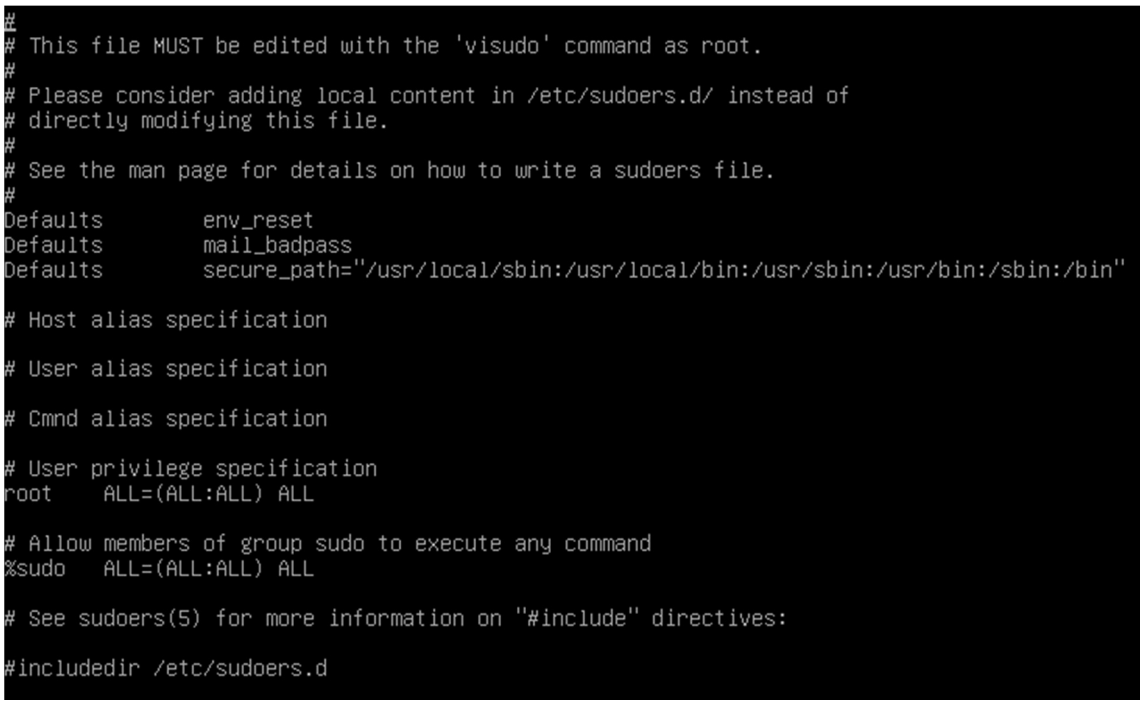
Para instalar o sudo simplesmente use o comando da Figura 12 em sistemas GNU/Linux Debian:

Figura 12 – Instalando o sudo

Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Depois de instalado é preciso configurar o arquivo /etc/sudoers, que é onde são parametrizados quais usuários podem ter acesso aos comandos definidos. Na Figura 13 a listagem default do mesmo.

Figura 13 – Listagem do /etc/sudoers



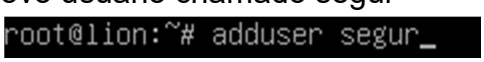
Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Como se pode observar, o usuário root tem permissão total [root ALL=(ALL:ALL) ALL] e usuários pertencentes ao grupo sudo [%sudo ALL=(ALL: ALL) ALL], também tem permissão total. (TURNBULL, 2005, p. 40).

Dependendo da forma como é a politica de segurança da empresa pode-se optar na criação de um usuário para cada admin e inserir suas permissões no arquivo, para cada aplicativo separadamente ou incuir o usuário no grupo sudo e permetir executar qualquer comando como super user.

Por exemplo, podemos incluir um novo usuário no sistema.

Figura 14 – Incluindo novo usuário chamado segur



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

E incluir permissões ao mesmo no arquivo /etc/sudoers, executando com o usuário root o comando visudo. Este comando apenas abre um editor já com o arquivo de configurações, onde digitamos as credenciais dos usuários, neste caso as seguintes.

Figura 15 – Incluindo credenciais para o usuário segur

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
segur   ALL=/sbin/ifconfig, /sbin/iptables
```

Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Assim o usuário segur vai poder executar os comandos *ifconfig* e *iptables* como se fosse o usuário dono dos mesmos, apenas adicionando o comando sudo na frente do mesmo, por exemplo:

Figura 16 – Executando comando como se fosse o root usando sudo

```
segur@lion:~$ sudo ifconfig
[sudo] password for segur:
eth0      Link encap:Ethernet  HWaddr [redacted]
          inet addr:[redacted]  Bcast:[redacted]  Mask:[redacted]
          inet6 addr: [redacted] Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5684 errors:0 dropped:0 overruns:0 frame:0
          TX packets:687 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2568256 (2.4 MiB)  TX bytes:55602 (54.2 KiB)
```

Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Note que antes de executar o comando, por segurança é solicitado a senha do usuário, pode-se opcionalmente desligar este recurso, mas não é recomendável. Além disso a senha será solicitada apenas no primeiro uso do sudo, nos comandos subsequentes, na mesma seção do usuário, não será mais solicitado.

Caso o usuário administrador precise ter privilégios idênticos ao de root e, dependendo da política de segurança da empresa, pode-se optar na inclusão do usuário no grupo sudo, assim, de cordo com as regras do arquivo sudoers, mostrado anteriormente, o usuário vai poder executar qualquer comando utilizando sudo. Isso evitaria de incluir uma regra para cada comando necessário para o usuário executar.

Aqui é importante lembra que sempre é possivel identificar todo movimento de uso do sudo nos logs do sistema, tais como no /var/log/secure ou /var/log/auth.log.

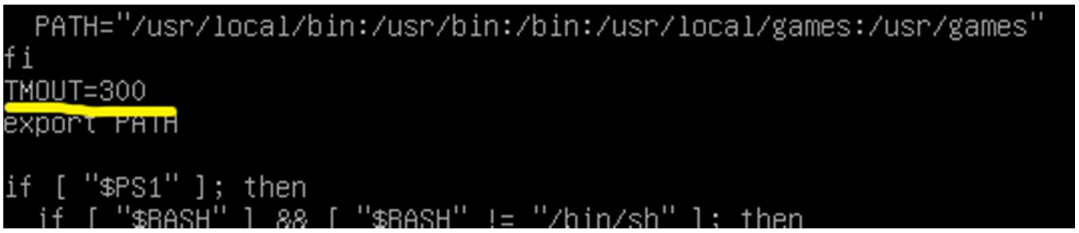
2.6 Segurança no terminal

A nível de terminal é preciso se preocupar não somente com a segurança provenientes de ataques remotos, assim como de funcionários mal intencionados que podem ter acesso físico ao local do servidor.

Quando um terminal de comandos está ativo com o usuário root, por exemplo, é importante terminar a seção após um tempo de inatividade. No sistema Linux, temos uma variável que controla o tempo de vida de um terminal chamada TMOUT, que não vem setada por padrão. (MELO, 2014, p. 20).

Seu uso é muito simples, no Debian usado no HCPA, simplesmente é acrescentado no arquivo /etc/profile o tempo que desejamos até o logout por inatividade no terminal.

Figura 17 – Incluindo TMOUT dentro do /etc/profile



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

No exemplo foi acresentado o tempo de 300 segundos, ou 5 minutos de inatividade para fazer logout do terminal. Como a configuração está no /etc/profile ela vai atuar em todos usuários que estiverem cadastarados no sistema.

Evitar o CTRL+ALT+DEL acidental também é uma boa prática para não reiniciar o servidor de forma desncessária.

No caso do Linux Debian 8, a configuração é um pouco mais complexa, pois não utiliza mais as configurações de inicialização em um único arquivo do SystemV (/etc/inittab), usando um arquivo para cada item dentro do diretório (/etc/init.d).

Este novo sistema de inicialização é controlado pelo *systemd* que é responsável por inicializar o espaço do usuário e gerenciar todos processos

posteriormente tal como registro de eventos e pode até mesmo substituir o *syslog*.

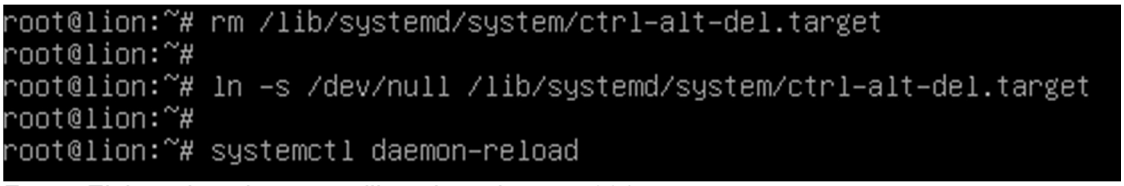
No *systemd* o comportamento do CTRL+ALT+DEL é controlado por /lib/systemd/system/ctrl-alt-del.target, na Figura 18 aparece na listagem do diretório.

Figura 18 – Listando o controle do CTRL+ALT+DEL

Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Então para mudar seu comportamento removemos o link simbólico que aponta para /lib/systemd/system/reboot.target e o fazemos apontar para nulo, /dev/null e após recarregamos o processo do serviço.

Figura 19 – Trocando o comportamento do CTRL+ALT+DEL



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

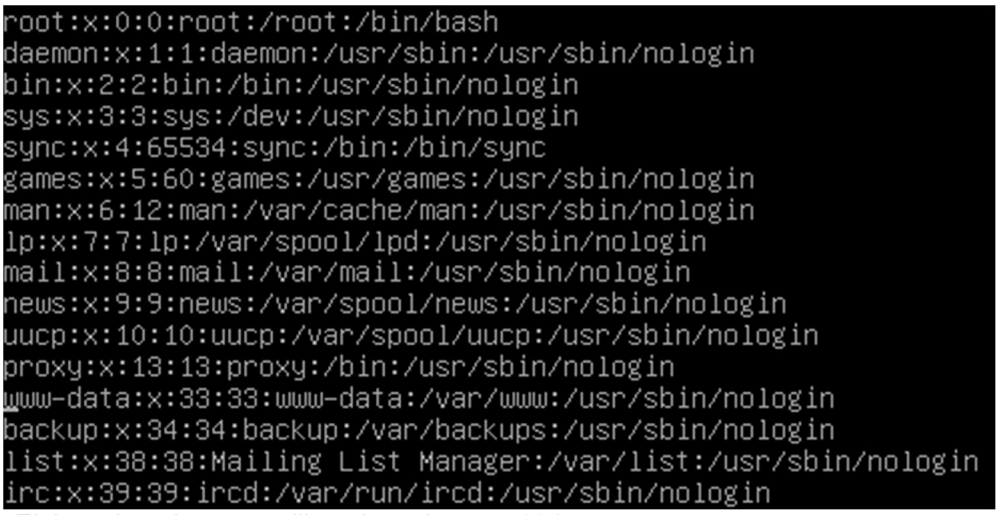
Este mecanimo do *systemd* para desativar o CTRL+ALT+DEL não está muito bem documentado e foi possível mudar seu comportamente atavés de consultas ao fórum oficial do Debian. (PEÑA, 2012, passim).

2.7 Manter shell para usuários que realmente precisa

Para habilitar a linha de comandos, shell do Linux, é preciso indicar no arquivo /etc/passwd para cada usuário do sistema. Alguns usuários de serviços, como por exemplo, no caso do Debian, o www-data é utilizado pelo serviço do servidor web Apache e não necessita fazer logon no sistema e nem precisa de shell para suas tarefas. (MELO, 2014, p. 112).

Na figura 20, é mostrado o conteúdo de um trecho deste arquivo para análise.

Figura 20 – Arquivo /etc/passwd



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Analisando o arquivo é possivel constatar que cada linha inicia com o nome do usuário e finaliza com o caminho do shell que será usado pelo mesmo.

No caso do root é indicado que ao fazer logon será executao o shell /bin/bash e no caso do www-data /usr/sbin/nologin, ou seja, este usuário não poderá fazer login no sistema, por questões de segurança.

No caso específico do servidor analisado, nenhuma alteração foi necessária, mas é importante poder certificar que somente os usuários que precisam de logon devam ter o shell indicado.

2.8 Segurança na administração remota

É muito comum administradores de sistema necessitar de acesso remoto para suas tarefas. Mesmo que o acesso so servidor seja feita somente dentro da rede local da empresa, que é o caso do Hospital de Clínicas de Porto Alegre - HCPA, alguns cuidados devem ser necessários.

Primeiramente é preciso deixar de usar qualquer serviço que transmite informações, incluindo senhas, em texto aberto pela rede, dentre eles podemos citar: telnet, ftp, rcp, rlogin e rsh. Qualquer um que estiver escutando a rede monitoriando o tráfego de dados poderá ter acesso as informações e usar para invadir o sistema. (TURNBULL, 2005, p. 169).

O acesso remoto ao servidor web, quando necessário, deve ser efetivado utilizando tráfego criptografado, através de uma conexão segura. Os servidores web analisados já possuem a ferramenta OpenSSH, que é um dos melhores pacotes livres para conexão remota.

O OpenSSH é uma suite de ferramentas que inclui o *ssh*, que substitui o *telnet* e *rlogin*, *scp* que substitui *rcp* e *sftp* uma modificação segura do *ftp*. Ele também contém o *sshd*, que é um SSH server, e *ssh-agent*, *ssh-keygen* e *ssh-add,* que manipula a geração e gerenciamento de chaves para o OpenSSH. Com ele também é possível obter conexões seguras através de tunelamento, que é uma forma de conexão criptografada. (TURNBULL, 2005, p. 170).

O SSH é a forma que podemos usar para conectar remotamente ao servidor e seu arquivo de configuração normalmente está localicado em: /etc/ssh/sshd_config.

Uma das ações pimordiais quando se fala em hardening do SSH normalmente envolve proibir o login remoto do super usuário. Para isso exite uma diretiva que pode ser incluída nos arquivos de configuração como mostrado no Quadro 6.

Quadro 6 - Diretiva para impedir logon remoto do super-usuário

```
PermitRootLogin no
```

Fonte: MELO (2014, p. 103); PEÑA (2012, cap. 5)

Forçar o uso da versão 2 do protocolo, para evitar falhas de segurança da versão 1, Quadro 7.

Quadro 7 - Forçar uso da versão segura do protocolo

```
Protocol 2
```

Fonte: MELO (2014, p. 103); PEÑA (2012, cap. 5)

Impedir que senhas em branco sejam aceitas, Quadro 8.

Quadro 8 - Impedir senhas em branco

```
PermitEmptyPasswords no
```

Fonte: MELO (2014, p. 104); PEÑA (2012, cap. 5)

Exitem outras configurações possíveis no SSH tal como forçar o uso de chaves simétricas (chave pública e privada), caso queira usar um sistema de autenticação diferente do padrão que é por senhas, mas no caso do HCPA não é usado.

2.9 Proteger contra ataque de força bruta

Apesar de a rede do HCPA já ser protegida por servidores de Firewall com proteções de IPS além de *modsecurity* com *Open Source Web Application Firewall*, podemos incluir uma proteção adicional no servidor web usando o software *Fail2ban*.

A sua instalação pode ser vista no Quadro 9:

Quadro 9 – Instalando fail2ban

```
sudo apt-get install -y fail2ban
```

Fonte: MELO (2014, p. 106).

Dois arquivos de configurações são disponibilizados:

- /etc/fail2ban/jail.conf
- /etc/fail2ban/fail2ban.conf

No fail2ban.conf é setado as configurações referente ao registro de eventos, no Quadro 10 é possível ver algumas configurações.

Quadro 10 – listagem do arquivo fail2ban.conf

```
[Definition]
loglevel = 3
logtarget = /var/log/fail2ban.log
socket = /var/run/fail2ban/fail2ban.sock
```

Fonte: MELO (2014, p. 106).

O arquivo jail.conf é definido em seções, onde a primeira denominada “default” é onde fica parametrizado o funcionamento do fail2ban através de algumas opções: (MELO, 2014, p. 106)

- **Ignoreip**: onde são definidos os IPs que não vão ser bloqueados pelo programa, ou seja, uma whitelist;
- **bantime**: onde é definido o tempo em segundos em que o IP ficará banido ou bloqueado;
- **maxretry**: define o número máximo em que o IP pode tentar efetivar um processo de login no servidor SSH até ser bloqueado;
- **logpath**: define o arquivo de log onde será registrada as tentativas de login que falharam;
- **destemail**: define o e-mail para o qual deverão ser encaminhadas as notificações de bloqueio;
- **banaction**: define qual ação será tomada.

Na segunda sessão é onde definimos quais serviços podem ser protegidos, tanto com o propósito de mitigação contra ataques de força bruta, como também negação de serviço. Abaixo uma relação de alguns serviços que já vem com predefinições de proteção ativados:

- Servidor web apache;
- Servidor SSH;
- Servidor de correio;
- Servidor de DNS.

3.0 Proteção serviço web

O Apache é o serviço web utilizado pelo servidor e por padrão ele retorna resposta HTTP contendo muitas informações, como a versão do Apache, PHP e até mesmo do Sistema Operacional. Como boa prática é importante fornecer o menor número de informações possíveis sobre versões de serviço ou que tipo de implemantação o servidor possui.

Baseado no conceito de segurança por obscuridade, é possível configurar o Apache para ocultar informações sobre o sistema. Essas informações surgem quando uma tela de erro ocorre. Para ocultar as informações do sistema, é

necessário editar o arquivo /etc/apache2/conf-enabled/security.conf, como demonstra o Quadro 11.

Quadro 11 – Configurações para Apache

De	Para
ServerToken Full	ServerToken Prod
ServerSignature on	ServerSignature off
TraceEnable on	TraceEnable off

Fonte: MELO (2014, p. 117).

- **ServerToken**: responsável por controlar o campo de cabeçalho de resposta do servidor. Deixando como Prod informa somente que o servidor é Apache, sem informar versão, ou tipo de Sistema Operacional, ou se tem módulo PHP ativado.
- **ServerSignature**: no modo off retira a linha de rodapé que poderia conter informações de sistema repassados ao usuário.
- **TraceEnable**: parâmetro usado apenas para teste e disgnósticos, em um server de produção pode ficar desativado.

Por padrão, em diretórios que não possuem nenhum arquivo index, o Apache lista todos os arquivos existentes nos mesmos. Normalmente este comportamento não é desejado, seja porque o diretório contém arquivos privados, ou apenas para não permitir que os arquivos que ali se encontram sejam vistos.

Para previnir este comportamento nos arquivos de configuração do Apache encontrados em /etc/apache2/apache2.conf é inserida a linhas do Quadro 12, na diretiva correspondente ao diretório home do site, que neste caso é /var/www.

Quadro 12 – Impedir listagem do conteúdo do diretório

```
Substituir
Options Indexes FollowSymLinks
Por
Options -Indexes +FollowSymLinks
```

Fonte: OURABI (2006, passim).

Outra questão importante é desabilitar as mensagens de erro na tela para os scripts PHP, pois o mesmo pode conter informações tais como nome do script em execução ou até mesmo detalhes da versão e conexões ao banco de dados. As configurações para o GNU/Linux Debian 8 normalmente se encontram no arquivo /etc/php5/apache2/php.ini, no Quadro 13 tem algumas diretivas importantes.

Quadro 13 – Opções desabilitadas no PHP.ini

```
#desabilitar mensagens gerais debug – opcional
display_errors = Off
register_globals = Off

# ativar modo de segurança sql
sql.safe_mode = Off
```

Fonte: OURABI (2006, passim).

O hardening no PHP não se resume apenas em desabilitar recursos, aqui é muito importante que os desenvolvedores elaborem códigos seguros, que possuam proteção para SQL Injection e outras ameaças tais como XSS (Cross-site scripting)

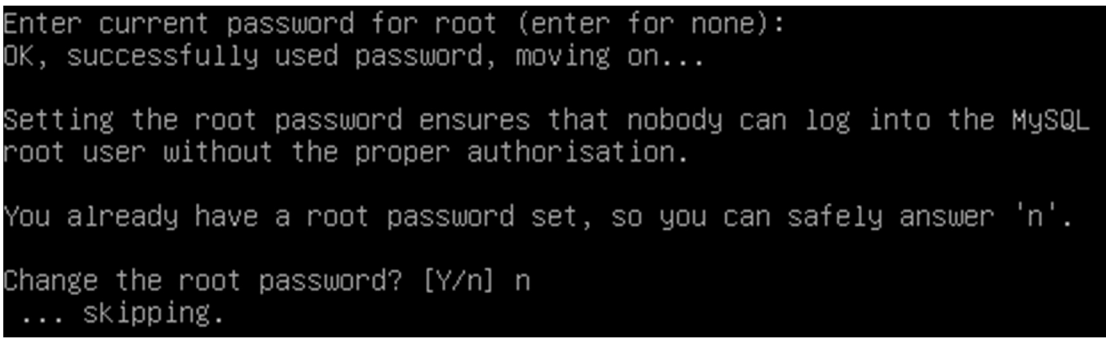
“É preciso que os programadores busquem informações sobre desenvolvimento usando boas práticas de segurança e customizações de segurança do seu ambiente PHP”. (MELO, 2014, p. 119)

3.1 Hardening no MySQL

Em todas plataformas, as distribuições MySQL incluem a linha de comando: *mysql_secure_installation*, que é um utilitário que automatiza alguns dos processos de segurança mais comuns de uma instalação deste banco de dados. Ele oferece a possibilidade de trocar a senha padrão da conta root, dentre outras coisas. O uso do comando é bem simples, pois é feito algumas perguntas ao usuário, que através de respostas simples, tal como sim ou não, é possível configurar o ambiente. (SECURITY in MySQL, 2017, p. 32).

A figura 21 exibe a primeira pergunta após executar o comando, esta talvez seja uma das conifurações mais importantes, onde a senha do usuário root do MySQL é alterada. No caso do HCPA a senha do mesmo já havia sido alterada, de forma que foi possível passar ao próximo passo.

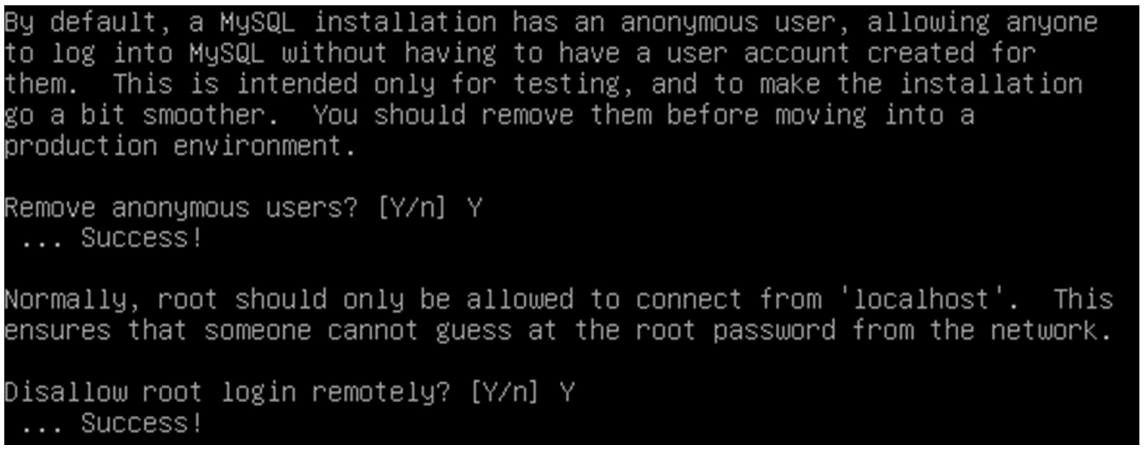
Figura 21 – Alterando a senha do root no MySQL



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Na seqüência, figura 22, o comando pergunta se queremos remover o usuário anonimo do sistema, e respondemos “Y”, *Yes*, para sim, afinal nosso banco de dados vai possuir apenas usuários autorizados. Logo após escolhemos sim também para que o usuário root possa conectar no banco somente em sessões locais, isso evita acessos remotos indevidos.

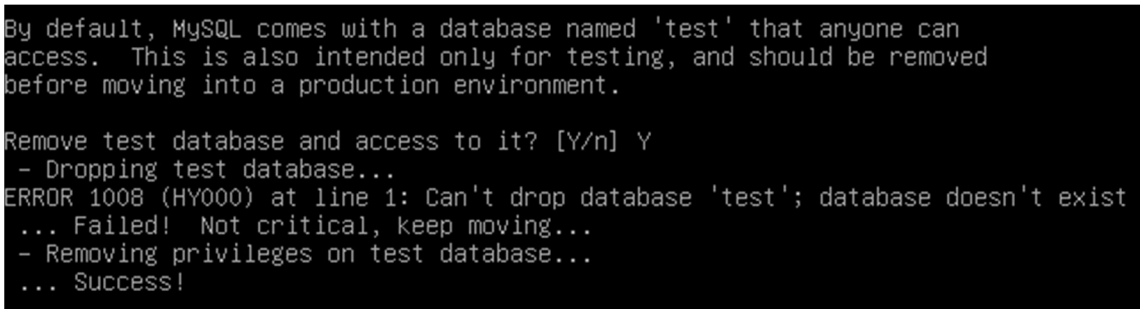
Figura 22 – Removendo usuário anonimo do MySQL



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Na figura 23, escolhemos sim para remover o banco de dados de teste que por default vem na instalação do MySQL, no caso do servidor web do HCPA já tinha sido removido.

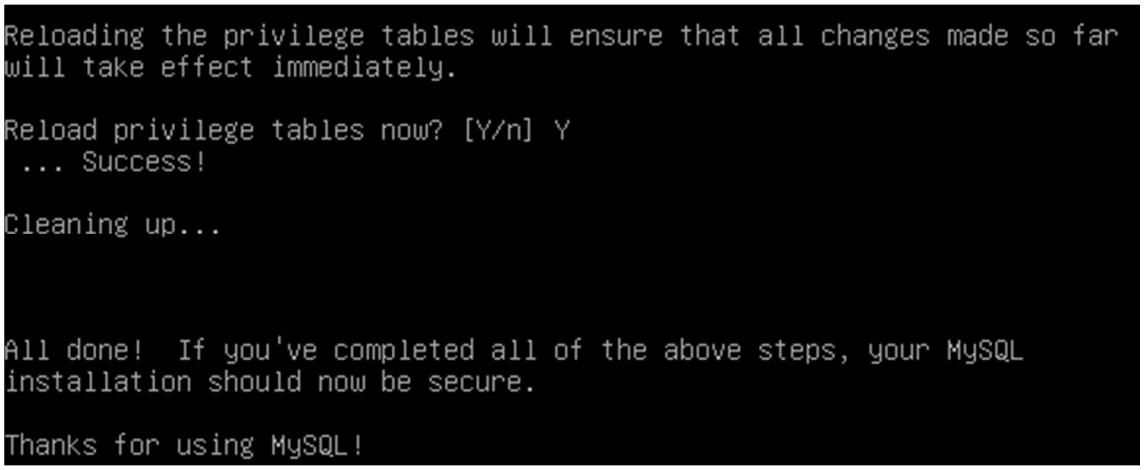
Figura 23 – Removendo banco de dados anonimo



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Por fim na figura 24, os privilégios dos usuários são recarregados para atualizar as permissões dos mesmos na sessão corrente.

Figura 24 – Atualizando privilégios e fim do hardening no MYSQL



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

3.2 Auditando as configurações do sistema

Ao final do processo de hardening é importante avaliar o nível de segurança atingido pelo sistema. Para isso será usado uma ferramenta de auditoria de segurança open source chamado Lynis. (CLÉMENT, 2017, p. 7).

A instalação da ferramenta é bem simples e pode ser executado diretamente no servidor analisado, sendo utilizado em várias tarefas relacionadas a segurança tais como:

- Checagem e auditoria da segurança do sistema;
- Avaliação da integridade dos arquivos;
- Sistema e arquivo forense;
- Serviço de relatórios e monitoramentos;
- Extensão para recursos de debug.

Na Figura 25 é mostrado a instalação do *Lynis*, no exemplo, o comando é executado no usuário root, caso seja com usuário diferente pode ser preciso preceder com o comando *sudo*, desde que o usuário tenha permissão nas configurações do sistema.

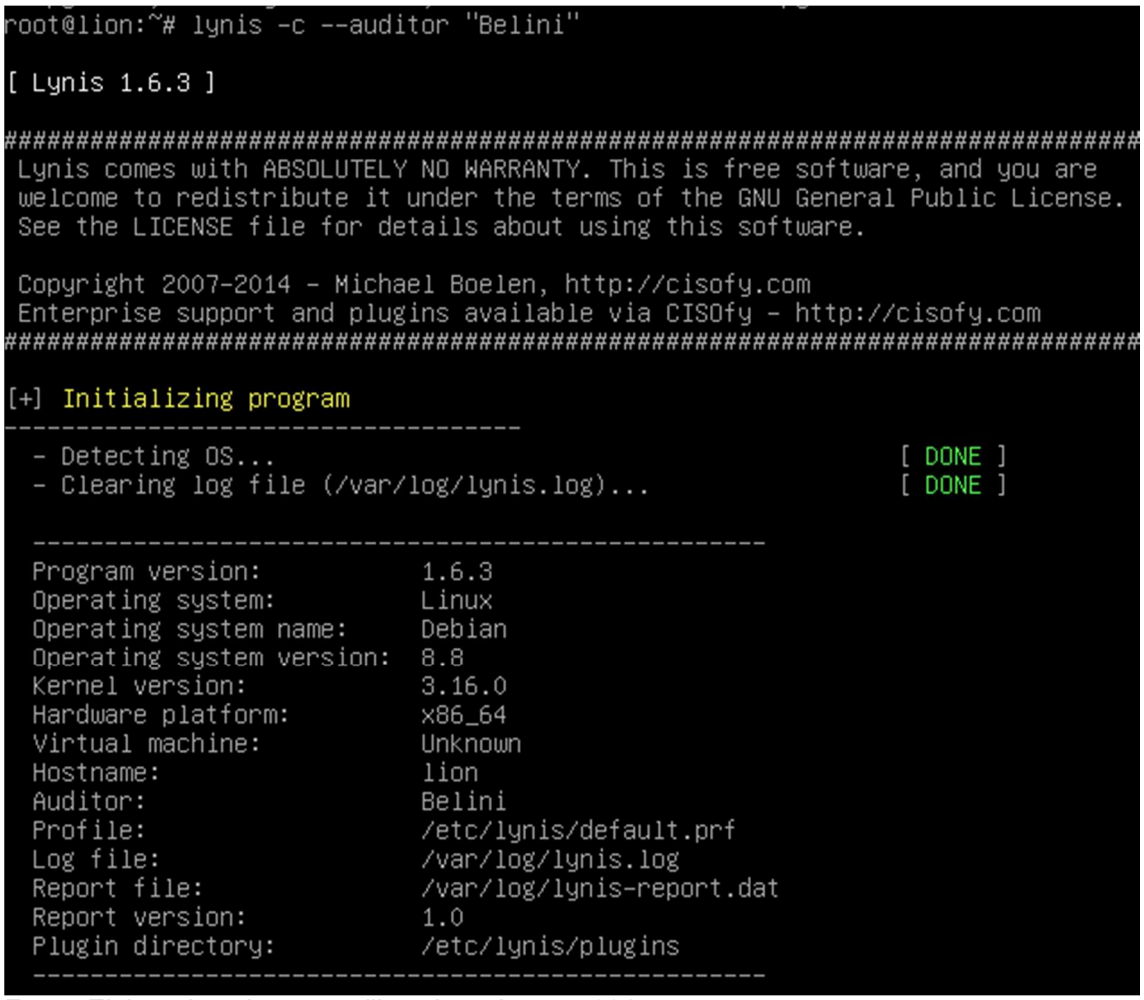
Figura 25 – Instalação do Lynis

Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Na Figura 26 o lynis é executado para auditar o sistema inteiro, também é mostrado o inicio da execução do mesmo:

- -c: analisa tudo (check-all)
- --auditor: Define o nome do auditor

Figura 26 – Inicio da execução do Lynis



Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Ao final são gerados dois arquivos de log no caminho /var/log:

- Teste e detalhamento: /var/log/lynis.log
- Relatório: /var/log/lynis-report.dat

Uma vez terminado a varredura é possível analisar os arquivos de log de diversas formas, pode se abrir em um editor de textos ou procurar por entradas “Warning” que precisam de atenção, Figura 27.

Figura 27 – Warnings gerados pelo Lynis no log

Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Ou até mesmo sugestões de melhoria, Figura 28.

Figura 28 – Sugestões de melhorias

Fonte: Elaborado pelo autor utilizando o sistema, 2017.

Através do conhecimento das configurações do sistema Linux e do estudo das melhorias sugeridas pelo Lynis é possível através de ciclos de análise de auditoria e configurações corretivas ou preventivas fortalecer a segurança do servidor através de uma melhoria continua.

4 CONCLUSÕES

A segurança da informação tem obtido cada vez mais importância, seja devido ao valor agregado onde a instituição depende cada vez mais dos sistemas informatizados ou pela ameaça de perda das informações ali contidas, ou mesmo interrupção de algum serviço importante.

Através da técnica de hardening, foi possível identificar que mesmo o sistema mais seguro ainda pode conter falhas de segurança e que o estudo do funcionamento e configuração das ferramentas utilizadas para prover serviços de informação devem sempre ser completamente analisados e verificados.

O avanço tecnológico constante leva os administradores de sistema a sempre procurar trazer a melhor da tecnologia para prover as necessidades dos

seus stakeholders, ou partes interessadas de um novo projeto ou serviço, mas é importante destacar que o uso de novas soluções podem abrir novas falhas de segurança. Cabe aos administradores de tecnologia da informação analisar e estar atentos na forma como as novas funcionalidades podem afetar a segurança e mitigar os riscos.

Mesmo em um sistema, aparentemente seguro, ainda é preciso monitorar, realizar testes de intrusão (Pentest), usar ferramentas de análise de configurações que ajudem a coletar dados que podem ser utilizados para melhoria dos sistemas informatizados.

Por fim, é importante lembrar que manutenção e melhoria continua, monitoramento e análise crítica, implementação e operação em conjunto com um bom planejamento forma um dos pilares da segurança da informação, onde tem no PDCA "Plan (Planejar), Do (Fazer), Check (Checar), Act (Agir)" a principal abordagem para implementação da segurança da informação. (ABNT, 2006, p. 4).

Em um trabalho futuro pode-se tentar agregar uma análise a nível de serviços web, submetendo o site para verificação em ferramentas de varredura de vulnerabilidades para certificar a segurança, usando protocolos seguros (SSL – Secure Socket Layer) e identificação de malware escondidos que podem comprometer a integridade de um portal web.

Abstract: This article presents a study to apply the technique of hardening, shielding of operating systems, in the web servers of Hospital de Clínicas of Porto Alegre - HCPA, with the purpose of increasing the security and in this way reducing the risks to the system. The work was done through a bibliographical research, seeking to obtain a script to follow, to identify and to fix fragile points in the security of a GNU / Linux system, used as a web server. To do this, the analysis tools and the hardenig techniques investigated were applied directly to the HCPA web server, presenting in the article the results obtained and the solutions performed. The research indicated several possible security threats to

occur in newly installed systems, or in active systems that are poorly configured, also showing several techniques to mitigate such risks, as well as the possibility of auditing the system through techniques and tools that show where an administrator should work to improve security on GNU / Linux web systems.

Keywords: security, servers, linux, kernel, lynis, php, mysql, hardening, hospital, hcpa

REFERÊNCIAS

ALESSANDRO, Silva. **Segurança em Servidores Linux.** Disponível em: <http://alessandrosilva.info/palestras/Seguran%C3%A7a_Linux.pdf >. Acesso em: 27 mai. 2017.

ABNT, Associação Brasileira de Normas Técnicas. **ABNT NBR ISO/IEC 27001**: Tecnologia da Informação – Técnicas de Segurança – Sistemas de gestão de segurança da informação – Requisitos. Rio de Janeiro, 2006.

BARBOSA, Felipe Santos. Fundamentos em Segurança e Hardening em Servidores Linux baseado na Norma ISO 27002. In: ENCONTRO UNIFICADO DE COMPUTAÇÃO EM PARNAÍBA, 5., 2012, Parnaíba. **Anais eletrônicos ENUCOMP 2012.** Disponível em: <https://drive.google.com/file/d/0B9j-X8nTcbJcSkxERnMzNU1oT0E/view?usp=sharing>. Acesso em: 15 jun. 2017.

BENTO, Juliano. **Hardening Linux 1º S.G.S.L / 5º F.G.S.L.**. Disponível em: <https://pt.slideshare.net/fgsl/palestra-hardening-linux-por-juliano-bento-v-fgsl-e-i-sgsl>. Acesso em: 20 abr. 2017.

CLÉMENT, Levallois, 2017. **Step-by-step guide to Linux security for beginners**. Disponível em: https://seinecle.github.io/linux-security-tutorials/generated-pdf/step-by-step-guide-en.pdf. Acesso em: 20 mai. 2017.

COSTA, Fábio. Hardening Linux. Disponível em: < https://www.pop-ba.rnp.br/pub/Site/Noticia0029/hardening1.pdf>. Acesso em: 27 mai. 2017.

DOBRZA´NSKI, Maciej. **Hardening MySQL**. Disponível em: <https://archive.fosdem.org/2013/schedule/event/hardening_mysql/attachments/slides/266/export/events/attachments/hardening_mysql/slides/266/hardening_mysql_security_fosdem_2013.pdf>. Acesso em: 22 jul 2017.

FONTES, Edison Luiz Gonçalves. **Segurança da informação: o usuário faz diferença**. São Paulo: Saraiva, 2006.

GRATTAFIORI, Aaron. **Understanding and Hardening Linux Containers**. [s.l]: NCC Group, 2016. Disponível em: <https://drive.google.com/file/d/0B9j-

X8nTcbJcM2l4RW9NS3ZKOWc/view?usp=sharing>. Acesso em: 20 mai. 2017.

MELO, Sandro. **Hardening em Linux**. Rio de Janeiro: Escola Superior de Redes, 2014. Disponível em: <https://mega.nz/#!kIEWgRjQ!VZkNd-16-LP5w8Uj2yiAm-q0nRez_oQYyPkmrYhK7xQ>. Acesso em: 25 jun. 2017.

NETO, Urubatan. **Dominando Linux Firewall IPTables**. Rio de Janeiro: Ciência Moderna Ltda., 2004.

OURABI, Yousef. **Hardening Linux Web Servers**, 2006. Disponível em: < http://freesoftwaremagazine.com/articles/hardening_linux/>. Acesso em: 10 jun. 2017.

PEÑA, Javier Fernández-Sanguino. **Security Debian Manual**, 2012. Disponível em: < https://www.debian.org/doc/manuals/securing-debian-howto/index.en.html>. Acesso em: 15 jun. 2017.

SECURITY in MySQL, 2017. Disponível em: <https://downloads.mysql.com/docs/mysql-security-excerpt-5.5-en.pdf>. Acesso em: 25 jul. 2017.

SILVA, Roberto Rodrigues, **Linux – Permissões de acessos especiais**. 2006. Disponível em: <https://www.vivaolinux.com.br/artigo/Linux-Permissoes-de-acesso-especiais>. Acesso em: 15 jun. 2017.

TIPPIT, INC, ITSECURITY. **10 Tips to Make Sure Your Firewall is Really Secure**. Disponível em: <https://drive.google.com/file/d/0B9j-X8nTcbJcWUlsOW1TeGQ4TUU/view?usp=sharing >. Acesso em: 11 jun. 2017.

THALHEIMER, Carsten. **MySQL Hardening**. Disponível em: <https://www.doag.org/formes/servlet/DocNavi?action=getFile&did=7535404>. Acesso em: 20 jul. 2017.

TURNBULL, James. **Hardening Linux**. New York: Springer-Verlag, 2005. Disponível em: <http://arsamandish.com/dl/ebook/Linux_eBooks_Collection/Apress%20Hardening%20Linux%20(2005).pdf>. Acesso em: 20 jun. 2017.