SECURITYDOG - SCRIPT DE HARDENING PARA SERVIDORES GNU/LINUX DEBIAN

Cairo Aparecido Campos
Antonio Eduardo Marques da Silva

RESUMO

O presente artigo apresenta o *script* SecurityDog para a blindagem de versões da distribuição GNU/Linux Debian, cuja o seu desenvolvimento teve como base a utilização de ferramentas e técnicas de segurança encontradas na bibliografia pesquisada. O *script* desenvolvido, facilita, automatiza e diminui o tempo gasto na realização de configurações de *hardening* de um servidor, além de servir como um roteiro de melhores práticas a ser seguido, para diminuir vulnerabilidades de um sistema após a sua instalação.

Palavras-chave: *Hardening*, Segurança da Informação, Debian, Servidor de Rede, *Shell script*.

ABSTRACT

The present article presents the script SecurityDog for shielding the versions of distribution of GNU/Linux Debian, whose its development had as foundation the use of tools and techniques of security found in the investigated bibliography. The developed script makes easier, automates and reduces the time spent in the realization of configurations of hardening of a server, besides serving as a guide of better practices to be followed, to reduce vulnerabilities of a system after its installation.

Keywords: Hardening, Information Security, Debian, Network Server, Shell script.

1 INTRODUÇÃO

Infelizmente muitos profissionais de TI inexperientes põe em produção servidores sem fazer as devidas configurações de segurança, o que possibilita que invasores explorem, sem muita dificuldade, suas vulnerabilidades.

Um servidor ao ser atacado ou invadido, além de ter suas informações roubadas pode servir de porta de entrada para a invasão de outros servidores e sistemas internos de uma empresa. A indisponibilidade de um servidor, o roubo de dados industriais e o comprometimento de um banco de dados, podem levar uma empresa a uma grande perda financeira e de competitividade em sua área de atuação ou mesmo levá-la a falência.

Para impedir a invasão de servidores, técnicas de *hardening* podem ser empregadas nestes equipamentos para protegê-los. Com o objetivo de fortalecer instalações do GNU/Linux Debian 9 e 10 em servidores, neste trabalho, após o estudo destas técnicas, foi desenvolvido o *script* SecurityDog, o qual é proposto para facilitar e automatizar as configurações de segurança.

Este trabalho está estruturado da seguinte forma. A primeira parte trata da presente introdução. A segunda parte traz o desenvolvimento do artigo que contem uma breve descrição sobre servidores de rede, uma introdução sobre a distribuição GNU/Linux Debian, uma contextualização sobre *hardening* e a descrição das técnicas de *hardening* implementadas no script desenvolvido. Na terceira e última etapa tem-se as considerações finais.

2. DESENVOLVIMENTO DO ARTIGO

2.1 Servidores de rede

Para Morimoto (2008), um servidor é uma máquina que trabalha de forma initerrupta para atender a requisições de máquinas clientes, por meio de “serviços” que nele operam (MORIMOTO, 2008). Computadores clientes ou nós, como também são chamados, são as estações de trabalho ou desktops utilizados por usuários para acessarem informações do servidor e que executam aplicações locais (FRANCISCATTO; CRISTO; PERLIN, 2014). Uma rede composta por servidores dedicados e computadores clientes é chamada de rede cliente-servidor (WILEY, 2011) e (FRANCISCATTO; CRISTO; PERLIN, 2014).

De acordo com Franciscatto, Cristo e Perlin (2014), um servidor deve possuir

hardware especifico (hardware de servidor) para que possa suportar um grande número de requisições, além de um sistema operacional que possibilite à máquina a capacidade de prover serviços de rede.

No que se refere ao sistema operacional utilizado, Morimoto (2008), esclarece que qualquer distribuição GNU/Linux pode ser usada em um servidor, pois os serviços utilizados (ex: Apache utilizado em servidores Web, Samba utilizado em servidores de arquivos, Bind utilizado em servidores DNS, MySQL utilizado em servidores de banco de dados, etc), são os mesmos, mudando apenas o processo de instalação. Porém, as distribuições mais usadas são o Debian, Ubuntu, CentOS, Fedora, Red Hat Enterprise e SuSE.

2.2 GNU/Linux Debian

2.2.1 Origem do Debian e versões ativas da distribuição

O Debian é uma distribuição GNU/Linux mantida pelo Projeto Debian, fundado por Ian Murdock em 16 de Agosto de 1993, o qual se tornou um grupo mundial de voluntários que desde o início esforçaram-se para produzir um sistema operacional composto inteiramente por software livre. O nome Debian originou-se da junção do nome do criador do projeto, Ian, com o nome de sua esposa, Debra, sendo que a pronúncia oficial é “débian” (GARBEE et al., 2002) e (GARBEE et al., 2017).

A distribuição Debian possui três versões em manutenção ativa, que são elas: *stable* (estável), *testing* (teste) e *unstable* (instável) (DEBIAN PROJECT, 2019a).

A versão *stable* é a última versão oficial lançada do Debian que é recomenda para aplicações que requerem estabilidade e segurança em nível de produção em servidores e para usuários novatos do GNU/Linux (DEBIAN PROJECT, 2019b). Quando uma nova versão *stable* é lançada, a versão *stable* anterior, que ainda recebe atualizações de segurança durante um ano da equipe de segurança do Debian, passa a ser chamada de *oldstable* (DEBIAN PROJECT, 2019b) e (DEBIAN PROJECT, 2019f). A versão *stable* atual é a versão 10 que possui o codinome Buster, lançada em 6 de julho de 2019 (DEBIAN PROJECT, 2019d) e a versão *oldstable* é a versão 9 de codinome Stretch (DEBIAN PROJECT, 2019e).

A versão *testing* contém pacotes que ainda não foram aceitos na versão *stable* mas que estão sendo avaliados ou “testados” para que isso ocorra futuramente (DEBIAN PROJECT, 2019a). Esta versão é recomendada para usuários avançados que desejam novos softwares em seu *desktop* e que são capazes de reportar e corrigir erros

para ajudar no desenvolvimento (DEBIAN PROJECT, 2019b). Recentemente, a nova versão *testing* recebeu o codinome Bullseye (DEBIAN PROJECT, 2019a) e (DEBIAN PROJECT, 2019b).

A versão *unstable* é onde o desenvolvimento ativo do Debian ocorre (DEBIAN PROJECT, 2019a). Ela é uma versão de desenvolvimento contínuo desta distribuição, que contém os pacotes mais recentes liberados pelos seus mantenedores (DEBIAN PROJECT, 2019c).

De forma geral, quando os pacotes de software do repositório da versão *testing* estão estáveis, ela é liberada como a nova versão *stable*, posteriormente um novo repositório *testing* é criado recebendo o próximo codinome planejado. Esta nova versão *testing* receberá pacotes da versão *unstable* como ocorria anteriormente, desde que estes não apresentem erros críticos de versão e suas dependências sejam satisfeitas por outros pacotes (DEBIAN PROJECT, 2019b) e (DEBIAN PROJECT, 2019c).

2.2.2 Codinomes de versão

Quando uma nova versão estável do Debian é lançada, ela possui um número de versão e o codinome herdado da versão *testing*. Os codinomes são baseados em personagens do filme da Pixar/Disney "*Toy Story"* (DEBIAN PROJECT, 2019b) e (DEBIAN PROJECT, 2019c). A primeira versão a possuir um codinome foi a Debian 1.1 "Buzz", lançada em 17 junho de 1996, devido a influencia de Bruce Perens que trabalhava na Pixar e que havia assumido a liderança do projeto Debian substituindo Ian Murdock (GARBEE et al., 2002), (GARBEE et al., 2017) e (DEBIAN PROJECT, 2019g).

Assim como as versões *testing* e *stable*, a versão *unstable* também possui um codinome, porém este sempre é mantido como o mesmo. O codinome Sid da versão *unstable*, faz referencia ao garoto malvado do filme que quebrava seus brinquedos e que criava criaturas estranhas usando suas peças (DEBIAN PROJECT, 2019b) e (DEBIAN PROJECT, 2019c).

Na Tabela 1, é listado os codinomes das versões *stable* lançadas, bem como número da versão e as datas de lançamento, fim de vida (*End Of Life* ou EOL), EOL LTS e ELTS EOL (DEBIAN PROJECT, 2019b).

O EOL LTS, no qual o LTS significa *Long Term Support* ou Suporte de Longo Prazo, tem como objetivo estender o tempo de vida de uma versão estável do Debian. Após a equipe de segurança do Debian não forcer mais atualizações para uma versão

oldstable, um grupo de volutários e companhias, formam uma equipe responsavel por fornecer atualizações de segurança durante dois anos. Deste modo uma versão *stable* consegue atingir cinco anos de vida (DEBIAN PROJECT, 2019h). Já ELTS EOL do qual ELTS significa *Extended Long Term Support* ou Suporte Estendido de Longo Prazo é uma oferta comercial para ampliar ainda mais o tempo de vida do Debian, após o fim do suporte do LTS. Ele é um projeto viabilizado por meio de patrocinadores e gerenciado pela Freexian, não sendo portanto um projeto oficial do Debian. Porém, mesmo que os patrocinadores decidam os pacotes que serão suportados, atualizações e correções de segurança estão disponíveis para todos usuários da distribuição sem qualquer custo (DEBIAN PROJECT, 2019i).

Tabela 1 – Versões stable do GNU/Linux Debian

Versão	**Codinome**	**Data de lançamento (Release date)**	**Data de fim de vida (End of life date)**	**EOL LTS**	**ELTS EOL**
12	Bookworm				
11	Bullseye				
10	Buster	06/07/2019	~2022		
9	Stretch	17/06/2017	~2020	~2022	
8	Jessie	25/04/2015	17/06/2018	~30/06/2020	
7	Wheezy	04/05/2013	25/04/2016	31/05/2018	~01/05/2019
6.0	Squeeze	06/02/2011	31/05/2014	29/02/2016	
5.0	Lenny	14/02/2009	06/02/2012		
4.0	Etch	08/04/2007	15/02/2010		
3.1	Sarge	06/06/2005	31/03/2008		
3.0	Woody	19/07/2002	30/06/2006		
2.2	Potato	15/08/2000	30/06/2003		
2.1	Slink	09/03/1999	30/09/2000	30/10/2000	
2.0	Hamm	24/07/1998			
1.3	Bo	02/07/1997			
1.2	Rex	12/12/1996			
1.1	Buzz	17/06/1996			
0.93R6		26/10/1995			
0.93R5		~01/03/1995			
0.91		~01/01/1994			

Fonte: Debian Project (2019b)

2.3 Hardening

2.3.1 Definição

Administradores com pouco conhecimento em segurança preparam seus servidores com uma instalação básica e depois que suas aplicações estão em funcionamento nada mais é feito para manter a integridade do sistema, pois existe a possibilidade de que uma aplicação pare de funcionar ao ser realizado um procedimento de segurança. Existe também, uma certa ilusão, na qual estes administradores julgam que, pelo fato do servidor e serviços nele instalados estarem em operação, já é o necessário, assim, as devidas técnicas de *hardening* não são aplicadas (REIS; VERBENA; JULIO, 2011), (BARBOSA, 2012) e (MELO, 2014).

A palavra "*hardening*" tem origem no idioma inglês e significa "endurecimento". No contexto de segurança computacional é o processo de proteger um sistema por meio da redução de suas possíveis vulnerabilidades, ao serem realizadas configurações que implementem controles específicos (MELO, 2014).

Ainda de acordo com Melo (2014), ao considerar a tradução do termo, endurecer um sistema operacional é o que deve ser feito antes de colocar um servidor em produção. Porém, o conceito de *hardening*, fica mais claro se a tradução for considerada como "foratalecimento", que pode ser explicado de forma empírica como um conjunto de configurações e melhorias, ou ainda, ajustes finos que vão gerar controles para que o sistema se torne mais seguro.

Para Reis, Verbena e Julio (2011), *hardening* ou blindagem de sistemas, como também é chamado, consiste na utilização de técnicas ou diretivas de segurança que devem ser seguidas antes, durante e após a instalação e configuração de servidores GNU/Linux. Estas técnicas possuem a finalidade de prover uma maior segurança tanto para servidores que estão expostos na internet ao disponibilizarem serviços externos, como servidores *Web*, quanto para servidores que disponibilizam serviços internos para o funcionamento de atividades empresariais, como servidores de banco de dados e de arquivos.

De acordo com System... (2019), o objetivo do *hardening* é eliminar tantos riscos de segurança quanto possível, ao reduzir a superfície de ataque de um sistema. Portanto, o primeiro passo a ser seguido para fortalecer um servidor GNU/Linux é definir a função do servidor e com isso determinar os serviços que precisam ser instalados. Instalar softwares ou executar serviços extras sem uma real necessidade,

possibilita a existência de vulnerabilidades que podem ser exploradas.

Ao ser iniciada a implementação das técnicas de *hardening*, há três fatores que devem ser levados em consideração: segurança, risco e flexibilidade. O analista de segurança ou administrador de redes deve dosar bem esses três fatores para definir um conjunto de controles que possam proporcionar ao sistema um equilíbrio entre produtividade e segurança. Quanto mais seguro for o sistema, menores serão os riscos e a flexibilidade. Porém, se o nível de flexibilidade for maior, a segurança diminui e o risco aumenta (REIS; JULIO, 2010), (REIS; VERBENA; JULIO, 2011), (BARBOSA, 2012), (MELO, 2014) e (MELLO, 2017).

Fazer uma instalação mínima do sistema operacional, criar partições de disco separadas que possuam as devidas permissões, remover pacotes de software não utilizados, instalar atualizações de segurança, desativar *shell* de contas utilizadas por serviços que não fazem login, remover permissão de `suid bit` de binários, instalar softwares de detecção de *`rootkits`*, definir o tempo de *timeout* de um *login* em um terminal, desabilitar o CTRL + ALT + DEL para evitar a reinicialização acidental ou maliciosa de um servidor e bloqueio do login do root no serviço SSH (Secure Shell), são alguns exemplos, entre muitos, de técnicas de *hardening* que podem ser utilizadas, as quais serão detalhadas a seguir.

2.3.2 Instalação e particionamento do sistema

Para a instalação do GNU/Linux Debian deve-se fazer o download de uma imagem de instalação (arquivo com a extensão ".iso"). Entre as opções de imagens disponíveis nos *mirrors* da distribuição, deve-se escolher a opção "netinst" que possibilitará fazer uma instalação mínima do sistema operacional (RIBEIRO, 2019).

Durante a instalação do sistema, uma prática recomendada para proporcionar uma maior segurança é particionar o disco e colocar os principais diretórios em partições separadas, o que possibilitará que cada partição possua sua tabela de alocação de arquivos, além de regras de montagem diferentes (REIS; JULIO, 2010) e (MELO, 2014).

Alguns autores sugerem separar os diretórios `/boot`, `/`, `/home`, `/usr`, `/tmp`, `/var`, `/var/log` em partições diferentes (BARBOSA, 2012), (MELO, 2014) e (MELLO, 2017). No entanto, de acordo com a nova versão *stable* do Debian, agora é recomendado manter o diretório `/usr` dentro da partição raiz "`/`" para evitar problemas de inicialização do sistema (PERENS et al., 2019). Um exemplo de partiocionamento

básico seria separar além da *swap*, os diretórios `/boot`, `/`, `/home`, `/tmp`, `/var`, `/var/log`.

Após o partiocionamento, na janela de "Seleção de software" é recomendado que nenhuma opção seja marcada para se ter uma instalação minima (RIBEIRO, 2019). Porém, caso posteriormente o servidor seja acessado de forma remota é aconselhavel que o serviço do SSH seja instalado nessa etapa marcando-se a opção "servidor ssh". Marcar a opção "utilitários de sistema padrão" instalará pacotes uteis para a administração do servidor, no entanto pacotes que devem ser removidos durante o *hardening*, como o cliente telnet e a ferramenta netcat também serão instalados. Após a instalação do sistema de acordo com novas necessidade, novos pacotes podem ser instalados, tendo-se assim sempre um sistema funcional com o minimo possivél de softwares instalados.

2.3.3 Criando contas para a administração

Após a instalação, novos logins devem ser criados para os usuários que irão administrar o servidor ao tornarem-se root. O exemplo abaixo adicionará três novos usuários:

```
adduser pedro
adduser maria
adduser jose
```

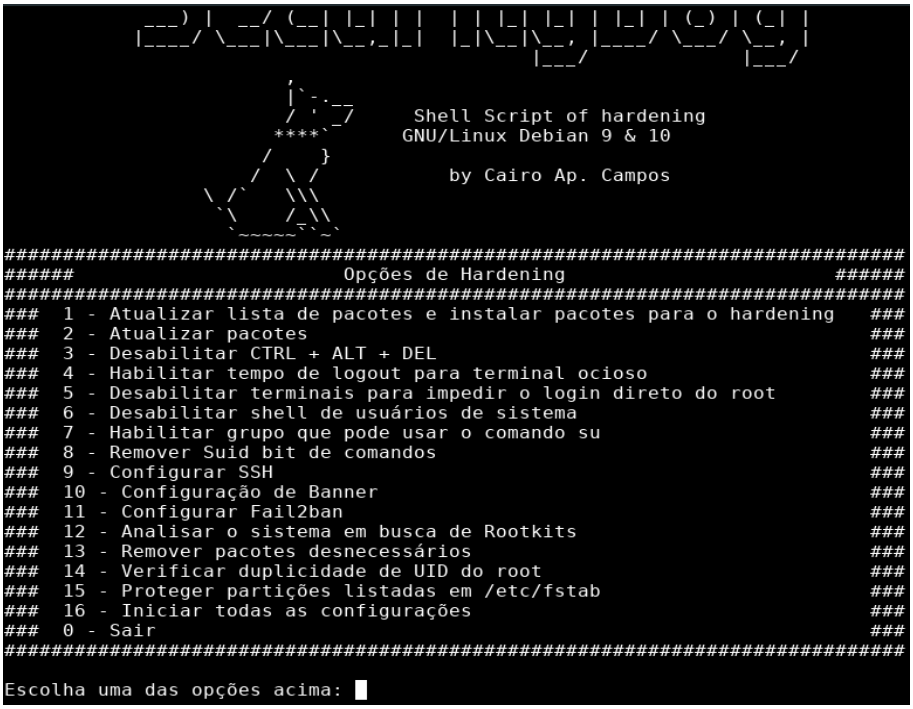
Caso um usuário não possa ter mais acesso, basta suspender sua conta usando o comando "`passwd -l login`".

2.3.4 Download do script de hardening

Após a criação dos *logins*, deve-se seguir os passos do tópico "Execução do *Script*" do repositório do GitHub[1] para fazer o download do SecurityDog e executá-lo. Ao iniciar o *script* terá-se o menu de opções da Figura 1.

[1] Link do repositório: https://github.com/cairoapcampos/SecurityDogV1

Figura 1 – Menu de opções do script SecurityDog



Fonte: autor (2019)

2.3.5 Atualizar lista de pacotes e instalar pacotes para o hardening

De acordo com a Figura 1, a primeira opção permite escolher os pacotes que podem ser instalados para o *hardening*. O administrador de sistema que queira todas as configurações pode instalar todos os pacotes de uma única vez ou instalar somente alguns pacotes de acordo com as configurações a serem realizadas.

Entre os pacotes disponiveis estão o `debsecan` (Debian Security Analyzer) que verifica a base de pacotes instalada, comparando-a com informações do registro de vulnerabilidades do CVE (*Common Vulnerabilities and Exposures*), que é um padrão mundial utilizado para enumerar vulnerabilidades conhecidas (MELO, 2014) e (MELLO, 2017).

Um segundo pacote que pode ser instalado é o `fail2ban` que é utilizado para conter, registrar e mitigar ataques de força bruta em um servidor SSH (MELO, 2014). Um terceiro pacote que pode ser instalado é o `rkhunter` que é uma ferramenta que verifica o sistema em busca de `rootkits`, `backdoors` e possíveis brechas locais. O `rkhunter` compara *hashes* SHA-1 de arquivos do sistema, com *hashes* de arquivos de um banco de dados *online*. Ao ser executada, esta ferramenta vasculha arquivos comumente usados pelos `rootkits`, permissões incorretas, arquivos escondidos e *strings* suspeitas em módulos do kernel (ALMEIDA, 2017).

O quarto e último pacote que pode ser instalado é o `htop` que é um visualizador de processos interativo, no qual é possivel rolar a lista na vertical e na horizontal para ver todos os processos e linhas de comandos completas, matar processos sem digitar o `pid`, matar vários processos de uma única vez e ajustar a prioridade de um processo sem digitar o seu `pid` e o valor de `renice` (MUHAMMAD, 2016).

2.3.6 Atualizar pacotes

Esta opção, ao ser executada, atualiza os pacotes não vulneraveis que possuem atualizações e ao utilizar o `debsecan` em conjunto com o `apt`, atualiza também os pacotes que possuem vulnerabilidades conhecidas e corrigidas. Além disso, ela também agenda automaticamente no `crontab` o script `DebsecanUpdatePkgs.sh` que atualizará o sistema automaticamente todo sábado a meia-noite.

Ao final da execução da opção ou do script `DebsecanUpdatePkgs.sh`, dois relatórios são criados em `/root/SecurityDogV1/Reports`. O primeio relatório chamado `VulnerableUpdatePkgs` possui a lista dos pacotes vulneráveis atualizados, no qual as informações nele gravadas para cada pacote contém o código CVE da vulnerabilidade, o nome do pacote e um resumo que indica o tipo da vulnerabilidade e a urgência da correção. O segundo relatório chamado `NormalUpdatePkgs` possui a lista dos pacotes atualizados que não possuem vulnerabilidades conhecidas.

2.3.7 Desabilitar CTRL + ALT + DEL

Na terceira opção de *hardening*, a combinação de teclas CTRL+ALT+DEL é desabilitada para evitar que o servidor seja reiniciado. De acordo com Melo (2014), esta configuração é uma boa prática principalmente quando o servidor GNU/Linux está no mesmo *rack* que também possui servidores que utilizam o sistema Microsoft Windows, assim caso a combinação de teclas sejam pressionadas equivocadamente, o sistema continuará a operar normalmente.

Para desabilitar o CTRL+ALT+DEL nas versões recentes do Debian que utilizam `systemd`, o comando `systemctl` mascara `ctrl-alt-del.target`, criando um *link* simbolico dele para `/dev/null`.

2.3.8 Habilitar tempo de logout para terminal ocioso

A quarta opção define o *logout* automático no terminal após um tempo de

inatividade. Para isto o tempo digitado em minutos durante o processo de hardening é convertido em segundos e atribuído a variável de ambiente `TMOUT` que é definida em `/etc/profile`.

2.3.9 Desabilitar terminais para impedir o login direto do root

Não é recomendável permitir que o usuário root efetue um *login* direto em um terminal local (`tty`). O correto é bloquear o *login* do root em todos os terminais, logar como usuário comum e, quando for necessário a realização de alguma tarefa administrativa, tornar-se usuário root usando o comando `su` (BARBOSA, 2012) e (MELO, 2014).

Para bloquear o *login* do root em terminais texto, a quinta opção do script comenta os terminais de `tty1` à `tty12` no arquivo `/etc/securetty`.

2.3.10 Desabilitar shell de usuários de sistema

Em sistemas GNU/Linux, existem três tipos de usuários. O primeiro tipo é o usuário root, que é o administrador do sistema. O segundo tipo refere-se aos usuários comuns que possuem uma senha para logar no sistema e acesso a uma pasta de usuário em `/home`. O terceiro e último tipo refere-se aos usuários de sistema que são responsáveis por controlar requisições de serviços (REIS; JULIO, 2010) e (REIS; VERBENA; JULIO, 2011).

O *shell* é uma interface que possibilita ao usuário digitar comandos e interagir com o sistema. Portanto, a sexta opção de *hardening* do *script* SecurityDog verifica no arquivo `/etc/passwd` os usuários de sistema que possuem um *shell* valido. Quando um shell valido (ex: `/bin/sh`) é encontrada em um desses usuários, ela é alterada para `/bin/false` com o comando `usermod`. Desta forma, reduze-se uma brecha que possibilitaria um acesso não autorizado ao sistema.

2.3.11 Habilitar grupo que pode usar o comando su

O PAM (*Pluggable Authentication Modules*), é um conjunto de bibliotecas compartilhadas responsável pela autenticação de usuários. Ele atua como um mediador entre as aplicações e a maneira como é feita a autenticação (STATO FILHO, 2016).

Ao utilizar a biblioteca `pam_wheel.so` é possível definir um grupo de usuários que poderão utilizar o comando `su`. Para isto, inicialmente, a sétima opção de *hardening* cria um grupo de acordo com o nome digitado pelo administrador de

sistemas e adiciona a ele usuários válidos que possuem uma pasta em `/home`. Posteriormente a linha comentada "`# auth required pam_wheel.so`" é alterada para "`auth required pam_wheel.so group=nomegrupo`" no arquivo `su` em `/etc/pam.d/`.

Para registrar o uso do comando `su` pelos usuários, a linha "`SULOG_FILE /var/log/sulog`" também é descomentada no arquivo `login.defs` em `/etc/` e o arquivo `sulog` que armazenará as informações é criado em `/var/log/`.

2.3.12 Remover suid bit de comandos

O `suid bit` é um tipo de permissão especial atribuída a binários. Quando um binário possui esta permissão é possível que um usuário o execute com os mesmos privilégios de seu dono. Caso o dono do binário seja o root, o usuário vai executar o binário como root (REIS; VERBENA; JULIO, 2011), (BARBOSA, 2012) e (MELO, 2014).

Muitos binários do sistema possuem a permissão de `suid bit`, pois alguns comandos podem, em alguma ocasião, ser utilizados por usuários comuns. Alguns exemplos de comandos que possuem a permissão de `suid bit` são o `su`, o `passwd` e o `mount` (BARBOSA, 2012), (MELO, 2014) e (MELLO, 2017).

Como uma medida de segurança, a oitava alternativa de *hardening* do SecurityDog, após fazer uma busca no sistema, remove a permissão de `suid bit` de todos os binários, exceto dos comandos `su` (usado por um usuário comum para tornar-se root) e `passwd` (utilizado por um usuário para trocar sua senha).

2.3.13 Configurar SSH

O SSH é uma ferramenta utilizada para acessar remotamente um servidor GNU/ Linux por meio de uma comunicação criptografada em uma rede. Porém, para uma maior segurança, detalhes importantes precisam ser levados em consideração ao configurar o arquivo `sshd_config` localizado em `/etc/ssh/` (REIS; VERBENA; JULIO, 2011).

Diante disto, a nona opção de *hardening,* quando executada, altera inicialmente no `sshd_config`, os itens `PermitRootLogin`, `PasswordAuthentication` e `PermitEmptyPasswords`. O item `PermitRootLogin` é definido como "`no`" para impedir que o usuário root faça login direto no SSH, o item `PasswordAuthentication` é definido como "`yes`" e o item

`PermitEmptyPasswords` é definido como "`no`", para impedir que contas de usuários sem senha façam *login*.

Posteriormente, o *script* permite alterar o item `port` para alterar a porta 22 padrão do ssh para o número de porta digitado, definir um grupo de usuários que poderão fazer *login* no serviço, ao inserir o item `AllowGroups` com o grupo criado durante o processo de *hardening* no arquivo `sshd_config` e definir um IP de uma interface de rede do servidor que pode ser utilizada para a conexão SSH.

Em um terceiro momento é possível configurar os arquivos do `TCP Wrappers`, `hosts.allow` e `hosts.deny` localizados em `/etc/`. No `hosts.allow` são definidos os IPs que podem conectar ao SSH para fazerem *login* e, no `hosts.deny`, os endereços IPs não liberados em `hosts.allow` são automaticamente broqueados.

2.3.14 Configuração de banner

Uma maneira de proteger *logins* SSH é exibir uma mensagem de repreensão para usuários não autorizados que tentam fazer *login* no servidor (SAIVE, 2012). Esta configuração, além de possuir uma função de advertência, possibilita também remover informações do arquivo `issue.net` em `/etc/` sobre a versão da distribuição GNU/Linux instalada, informação esta que pode ser utilizada por invasores para explorar alguma vulnerabilidade (MELO, 2014). Além do arquivo `issue.net`, o arquivo `motd` também exibe uma mensagem, porém esta é exibida após um usuário fazer *login* (SAIVE, 2012).

Em distribuições GNU/Linux Debian atuais, embora o arquivo `issue.net` exista, ele não é habilitado por padrão no SSH, o que impossibilita que uma mensagem seja exibida ao tentar-se fazer login.

Diante disso, a décima opção de *hardening* habilita o *banner* `issue.net` no arquivo `sshd_config` em `/etc/ssh/` e desabilita os arquivos `issue.net` e `motd` padrão, posteriormente novos arquivos `issue.net` e `motd` com mensagens em inglês ou português são definidos de acordo com a linguagem escolhida ao executar a opção. No *banner* do `motd` é definida uma mensagem de boas-vidas, levando-se em consideração que o usuário que autenticou é alguém que possui acesso autorizado.

2.3.15 Configurar Fail2ban

Como visto anteriormente o `Fail2ban` é um software para mitigar ataques de

força bruta. Para melhor utilizá-lo, a décima primeira opção de *hardening* possibilita alterar no arquivo `jail.local` localizado em `/etc/fail2ban`, as configurações dos itens `ignoreip`, `bantime`, `maxretry` e da porta do SSH que é monitorada. No `ignoreip` são definidos os IPs que não vão ser bloqueados pelo programa (`whitelist`), no `bantime` é definido o tempo em segundos em que o IP ficará banido e no `maxretry` é definido o número máximo de tentativas em que um IP pode tentar efetivar um processo de *login* no servidor ssh até ser bloqueado. No que se refere a porta do SSH que deve ser monitorada, o SecurityDog lê o arquivo `sshd_config` e define a mesma porta utilizada por ele.

2.3.16 Analisar o sistema em busca de Rootkits

A décima segunda opção de *hardening* altera o arquivo `rkhunter.conf` em `/etc/` e o arquivo `rkhunter` em `/etc/default/` para que atualizações de assinaturas de *rootkits* e propriedades de arquivos sejam realizadas, além de habilitar o escaneamento diário e automático que utiliza o *script* `rkhunter` localizado em `/etc/cron.daily/`.

Posteriormente, a versão instalada do `rkhunter` é verificada, as assinaturas de *rootkits* e propriedades dos arquivos são atualizadas e uma verificação do sistema é iniciada. Ao fim do escaneamento da verificação, as informações mostradas na tela também podem ser encontradas no arquivo de log `rkhunter.log` em `/var/log/` que também armazena as informações do escaneamento automático diário.

2.3.17 Remover pacotes desnecessários

Mesmo após a instalação básica do sistema, uma verificação minuciosa dos programas instalados, listados pelo comando `dpkg -l`, deve ser realizada e programas desnecessários devem ser removidos (REIS; VERBENA; JULIO, 2011).

A décima terceira opção de *hardening* remove pacotes relacionados a dispositivos *bluetooth* e *wireless*, além de remover os pacotes `netcat`, cliente `telnet` e o `git` que podem ser utilizados para a transferência de *scripts* e *malwares*. A remoção do comando `wget` também é recomenda por alguns autores como (REIS; JULIO, 2010), (REIS; VERBENA; JULIO, 2011), (BARBOSA, 2012) e (MELO, 2014). Porém este comando é utilizado pelo `rkhunter` para baixar arquivos que possuem informações sobre a versão instalada e assinaturas de `rootkits`, portanto ele é mantido.

2.3.18 Verificar duplicidade de UID do root

Apenas a conta de root deve ter `UID` com o valor de `0`, devido aos privilégios administrativos que ela possui (GITE, 2019). Por desconhecimento, um administrador inexperiente pode atribuir o `UID 0` para que um usuário comum tenha os mesmos privilégios do usuário root, abrindo assim uma brecha no sistema, pois este usuário caso seja descoberto por uma pessoa mal-intencionada, pode ser utilizado para acessar o servidor.

A décima quarta opção do *script* de *hardening* analisa o arquivo `passwd` em `/etc/` e lista, caso existam, usuários que, além do root, possuam o `UID` como `0`.

2.3.19 Proteger partições listadas em /etc/fstab

Como visto no tópico 2.3.2, o disco pode ser partionado em `/boot`, `/`, `/home`, `/tmp`, `/var`, `/var/log`, além da partição `swap`. A décima quinta opção de *hardening* edita o arquivo `fstab` em `/etc/` alterando as configurações de montagem das partições, para proteger o sistema de arquivos. A Tabela 2, lista as opções de montagem atribuídas para cada partição.

Tabela 2 – Opções de montagem de cada partição

Ponto de Montagem	nosuid	noexec	noatime	nodev
/boot	X	-	-	-
/	-	-	-	-
/home	X	X	-	X
/tmp	X	X	-	X
/var	X	X	-	X
/var/log	X	X	X	X

Fonte: adaptado de Barbosa (2012), Melo (2014) e Mello (2017)

Entre as opções de montagem de partição, o `nosuid` inibe que binários com permissão de `suid bit` sejam executados, o `noexec` impossibilita a execução de qualquer binário ou arquivo executável, o `nodev` tira o suporte de arquivos de dispositivos e o `noatime` ao estar parametrizado em uma partição, possibilita ao kernel deixar de executar uma rotina de atualização dos metadados de tempo de acesso, o que ajuda na performance do sistema (MELO, 2014) e (MELLO, 2017).

Embora as opções de montagem descritas acima sejam uma configuração de *hardening* interessante, ao se utilizar os comandos `apt`, `apt-get`, `aptitude` e

dpkg para a instação de novos pacotes de software, nada poderá ser instalado corretamente, pois existirão erros na gestão de pacotes. Estes erros acontecem devido aos comandos de gestão de pacotes executarem e gravarem informações nos diretórios "/var" e "/tmp" que estão parametrizados com a opção noexec. Para resolver este problema basta remontar as partições com permissão de execução com os comandos abaixo e instalar os pacotes (MELO, 2014):

```
mount –o remount,rw,exec /var
mount –o remount,rw,exec /tmp
```

Após a instalação dos pacotes, as partições devem ser remontadas com permissão de não execução com os comandos abaixo:

```
mount –o remount,rw,noexec /var
mount –o remount,rw,noexec /tmp
```

2.3.20 Iniciar todas as configurações

Como o próprio título sugere, esta opção do *script* inicia as opções de *hardening* de 1 a 15, caso o administrador queira aplicar todas as configurações.

3. CONSIDERAÇÕES FINAIS

Embora as configurações de *hardening* implementadas pelo *script* SecurityDog pareçam ser relativamente simples, existe um grande desconhecimento de administradores de redes, profissionais de segurança e gestores de TI sobre sua importância e como implementá-las.

O *script* desenvolvido possibilitou um roteiro de melhores práticas a ser seguido para o fortalecimento de uma distribuição GNU/Linux Debian após sua instalação. Para uma melhor eficiência, este roteiro deve ser utilizado em conjunto com outras práticas de segurança.

Entre estas práticas, destaca-se a separação de cada serviço de rede em uma VM (*Virtual Machine)* ou *container*, utilização de um servidor de log remoto, realização com frequência de *backups* de arquivos de configuração, bases de dados e de arquivos do negócio, instalação e configuração de um IDS (*Intrusion Detect System*) no datacenter, verificação de portas abertas de um servidor durante sua configuração inicial, realização de testes de invasão (*pentest*) para avaliar a infraestrutura e os

serviços implantados, atualização com intervenção do administrador de redes de aplicações `web` do tipo `CMS` (*Content Management System*) como `WordPress`, `Joomla` e `Drupal` que não são atualizadas por gerenciadores de pacotes do sistema e configuração do *firewall* local de cada servidor para atuar como uma camada adicional de segurança que estará abaixo do *firewall* central. Além disso, mas não menos importante, uma documentação sobre a instalação e configuração de serviços, aliada a um plano de continuidade de negócios, devem ser criados para atuarem como ferramentas que auxiliarão na recuperação da infraestrutura e serviços de TI, após desastres ou invasões que possam comprometer ou tornar indisponíveis os servidores.

Como trabalhos futuros, espera-se adicionar no *script* a opção de *hardening* `port knocking`, que também é utilizada para proteger o SSH, além de opções que possibilitem o fortalecimento de serviços `web` como o `Apache`, serviços de banco de dados como o `MySQL`, serviços `DNS` como o `BIND`, entre outros.

4. REFERÊNCIAS

ALMEIDA, R. Q. RKHUNTER - The Rootkit Hunter Project. 2017. Disponível em: <https://www.dicas-l.com.br/arquivo/rkhunter_the_rootkit_hunter_project.php>. Acesso em: 11 ago. 2019.

BARBOSA, F. S. Fundamentos em Segurança e Hardening em Servidores Linux baseado na Norma ISO 27002. In: ENCONTRO UNIFICADO DE COMPUTAÇÃO EM PARNAÍBA (ENUCOMP), 5., 2012, Parnaíba. **Anais Eletrônicos ENUCOMP 2012.** Parnaíba: Fuespi, 2012. p. 27 – 66. Disponível em: <https://pt.scribd.com/document/273102285/ENUCOMP-2012>. Acesso em: 03 ago. 2019.

DEBIAN PROJECT. **Debian Releases.** 2019a. Disponível em: <https://www.debian.org/releases/index.en.html>. Acesso em: 25 maio 2019.

_____. **DebianReleases.** 2019b. Disponível em: <https://wiki.debian.org/DebianReleases>. Acesso em: 25 maio 2019.

_____. **DebianUnstable.** 2019c. Disponível em: <https://wiki.debian.org/DebianReleases>. Acesso em: 25 maio 2019.

_____. **Debian 10 “buster” released.** 2019d. Disponível em: <https://www.debian.org/News/2019/20190706>. Acesso em: 25 maio 2019.

_____. **Debian Stretch.** 2019e. Disponível em: <https://wiki.debian.org/DebianStretch>. Acesso em: 25 maio 2019.

_____. **OldStable.** 2019f. Disponível em: <https://wiki.debian.org/DebianOldStable>. Acesso em: 25 maio 2019.

_____. **ToyStory.** 2019g. Disponível em: <https://wiki.debian.org/ToyStory>. Acesso em: 25 maio 2019.

_____. **Debian Long Term Support.** 2019h. Disponível em: <https://wiki.debian.org/pt/LTS>. Acesso em: 25 maio 2019.

_____. **Extended Long Term Support.** 2019i. Disponível em: <https://wiki.debian.org/LTS/Extended>. Acesso em: 25 maio 2019.

FRANCISCATTO, R.; CRISTO, F.; PERLIN, T. **Redes de computadores.** Frederico Westphalen: Universidade Federal de Santa Maria, Colégio Agrícola de Frederico Westphalen, 2014. 116 p. Disponível em: <https://www.ufsm.br/unidades-universitarias/ctism/cte/wp-content/uploads/sites/413/2018/12/redes_computadores.pdf>. Acesso em: 16 jul. 2019.

GARBEE, B. et al. **Uma Breve História do Debian.** 2002. Disponível em: <http://rotasul.net/linux/historia_debian.pdf>. Acesso em: 25 maio 2019.

_____._____. 2017. Disponível em: <https://www.debian.org/doc/manuals/project-history/project-history.pt.pdf>. Acesso em: 25 maio 2019.

GITE, V. **40 Linux Server Hardening Security Tips [2019 edition].** 2019. Disponível

em: <https://www.cyberciti.biz/tips/linux-security.html>. Acesso em: 13 ago. 2019.

MELO, S. **Hardening em Linux.** Rio de Janeiro: RNP/ESR, 2014. 278 p. Disponível em: <https://pt.scribd.com/doc/254117692/Hardening-em-Linux>. Acesso em: 03 ago. 2018.

MELLO, B. F. **Estudo da aplicação da técnica de Hardening nos servidores web do Hospital de Clínicas de Porto Alegre.** 2017. Disponível em: <https://riuni.unisul.br/bitstream/handle/12345/3164/Belini_Artigo_TCC_Versao_Final_Publicacao.pdf?sequence=1&isAllowed=y>. Acesso em: 03 ago. 2019.

MORIMOTO, C. E. **Servidores Linux:** Guia Prático. 2008. Disponível em: <https://www.hardware.com.br/livros/servidores-linux/servidores-linux.html>. Acesso em: 25 maio 2019.

MUHAMMAD, H. **Htop.** 2016. Disponível em: <https://github.com/hishamhm/htop>. Acesso em: 11 ago. 2019.

PERENS, Bruce et al. **Guia de Instalação de Debian GNU/Linux.** 2019. Disponível em: <https://www.debian.org/releases/stable/amd64/install.pdf.pt>. Acesso em: 10 ago. 2019.

REIS, F. A.; JULIO, E. P. **Hardening em Sistemas Operacionais GNU/LINUX.** 2010. Disponível em: <http://re.granbery.edu.br/artigos/Mzk3.pdf>. Acesso em: 03 ago. 2019.

REIS, F. A.; VERBENA, M. F.; JULIO, E. P. **Hardening.** 2011. Revista Infra Magazine - Edição 1. Disponível em: <https://www.devmedia.com.br/hardening-artigo-revista-infra-magazine-1/20818>. Acesso em: 03 ago. 2019.

RIBEIRO, Fernando. **Servidor Debian 10 'Buster'.** 2019. Disponível em: <https://servidordebian.org/pt/buster/start>. Acesso em: 10 ago. 2019.

SAIVE, R. **Protect SSH Logins with SSH & MOTD Banner Messages.** 2012. Disponível em: <https://www.tecmint.com/protect-ssh-logins-with-ssh-motd-banner-messages/>. Acesso em: 15 ago. 2019.

STATO FILHO, A. **Falando sobre o PAM (Pluggable Authentication Modules).** 2016. Disponível em: <https://stato.blog.br/wordpress/falando-sobro-o-pam-pluggable-authentication-modules/>. Acesso em: 12 ago. 2019.

SYSTEM Hardening Guide. 2019. Disponível em: <https://n0where.net/system-hardening-guide>. Acesso em: 03 ago. 2019.

Wiley, J. **Windows Server® Administration Fundamentals, Exam 98-365.** [s. L.]: Wiley, 2011.