POTENCIALIZANDO A DEFESA DE SERVIDORES WEB COM ESTRATÉGIAS DE HARDENING

ENHANCING WEB SERVER DEFENSE WITH HARDENING STRATEGIES

Evandro Ferreira Melo Pires, Faculdade de Tecnologia de Araraquara
evandrofemp@gmail.com

Leonardo Araújo dos Santos, Faculdade de Tecnologia de Araraquara
leonardo.arasantos@gmail.com

João Emmanuel D' Alkmin Neves, Mestre, Faculdade de Tecnologia de Americana
professoralkmin@gmail.com

Resumo

Este artigo ressalta a importância do hardening de servidores web como medida crucial para reforçar a segurança dos sistemas. O hardening envolve a implementação de configurações de segurança para reduzir vulnerabilidades e proteger contra ataques. Destacam-se a atualização regular de software para evitar brechas, a configuração adequada de servidores desabilitando serviços desnecessários e definindo permissões, e a aplicação de certificados SSL/TLS para segurança dos dados transmitidos. O gerenciamento de autenticação e controle de acesso é enfatizado, incluindo autenticação robusta, senhas fortes e restrições de acesso. São mencionadas medidas como firewalls e políticas de senha, bem como o monitoramento e registros para identificar atividades suspeitas. A ferramenta Nessus é citada para testes de vulnerabilidade e análise. O artigo propõe a adoção proativa de medidas, incluindo testes com o Nessus, para identificar e mitigar vulnerabilidades, protegendo dados e serviços contra ataques.

Palavras-chave: Hardening. Segurança Cibernética. Servidores Web. Mitigação. Acesso.

Abstract

This article highlights the significance of web server hardening as a critical measure to enhance system security. Hardening involves implementing security configurations to reduce vulnerabilities and defend against attacks. Key points include regular software updates to prevent breaches, proper server configuration by disabling unnecessary services and setting permissions, and the application of SSL/TLS certificates for secure data transmission. Authentication management and access control are emphasized, encompassing robust authentication, strong passwords, and access restrictions. Measures like firewalls and password policies are mentioned, along with monitoring and logging to identify suspicious activities. The Nessus tool is referenced for vulnerability testing and analysis. The article advocates for the proactive adoption of measures, including Nessus testing, to identify and mitigate vulnerabilities, safeguarding data and services against attacks.

Keywords: *Hardening, Cybersecurity, Web Servers, Mitigation, Access.*

1 INTRODUÇÃO

Com o avanço das tecnologias da informação e a crescente dependência da internet para diversas atividades, a segurança dos servidores web tornou-se uma preocupação crítica. Os servidores *web* são a espinha dorsal de muitos serviços e aplicações online, e sua vulnerabilidade pode resultar em consequências graves, como roubo de informações sensíveis, interrupção de serviços e comprometimento da integridade dos dados.

Como dito por Fagundes (2017 apud TURNBULL, 2005, p. 1) O termo em inglês "*hardening*", traduzido literalmente como "endurecimento", refere-se ao processo de fortificação de um sistema operacional por meio da aplicação de técnicas específicas de controle para melhorar suas configurações, com o objetivo de torná-lo mais seguro. No âmbito da segurança da informação, esse termo também significa proteger um sistema ao reduzir as suas vulnerabilidades.

Nesse contexto, o conceito de *hardening* para servidores *web* surge como uma abordagem essencial para fortalecer a segurança desses sistemas. O *hardening* consiste em implementar uma série de medidas e melhores práticas com o objetivo de reduzir as possibilidades de exploração de vulnerabilidades. Essas medidas abrangem desde a configuração adequada dos serviços a implementação de mecanismos de proteção e monitoramento até mesmo componentes do servidor.

Este artigo científico está alinhado com um dos dezessete principais objetivos de desenvolvimento sustentável global das Nações Unidas, que tem um apelo em contribuir com ações para acabar com pobreza, proteger o meio ambiente, o clima e garantir que as pessoas possam desfrutar da paz e prosperidade, que é o ODS “16.10 - Assegurar o acesso público à informação e proteger as liberdades fundamentais, com conformidade com a legislação nacional e os acordos internacionais (Nações Unidas Brasil, 2023).”

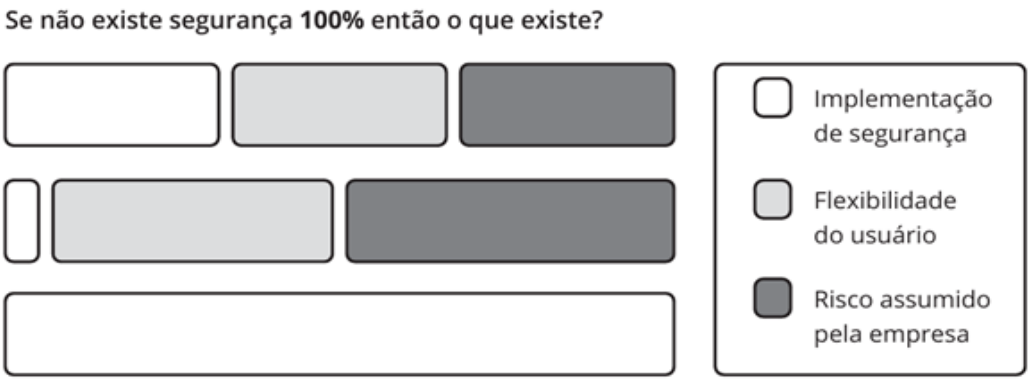
Algumas técnicas e estratégias de *hardening* abordam os aspectos relacionados à configuração segura dos servidores, gerenciamento de autenticação e controle de acesso, além de práticas para mitigar ataques comuns, como injeção de código, *cross-site scripting* e ataques de negação de serviço.

Além disso, será explorada uma ferramenta de segurança de redes conhecida como *Nessus* para auxiliar na análise de vulnerabilidades, análise de conformidade, que traz relatórios detalhados e escaneamento contínuo**.**

2 REFERENCIAL TEÓRICO

Compreender os desafios e as consequências de uma violação de segurança nos servidores web permite reconhecer a necessidade de implementar medidas proativas para mitigar riscos. No conceito de *hardening* existem três fatores a se pensar: segurança, risco e flexibilidade, como ilustrada Figura 1 abaixo.

Figura 1 Segurança, risco, flexibilidade



Fonte: MELO, Sandro. Hardening em Linux. Rio de Janeiro: Editora Rede Nacional de Ensino e Pesquisa – RNP, 2014.

Encontrar o equilíbrio ideal entre produtividade e segurança é um desafio que requer habilidade para equilibrar cuidadosamente esses três fatores. É necessário determinar um conjunto de controles que possa garantir um nível adequado de segurança ao sistema, sem comprometer significativamente a produtividade. Por tanto a configuração segura dos servidores *web* é fundamental para reduzir a superfície de ataque, desabilitando serviços desnecessários, aplicando atualizações de segurança e definindo permissões adequadas. A criptografia, por meio dos protocolos SSL/TLS, garante a confidencialidade e integridade dos dados transmitidos entre clientes e servidores.

2.1 ATUALIZAÇÕES DE SOFTWARES

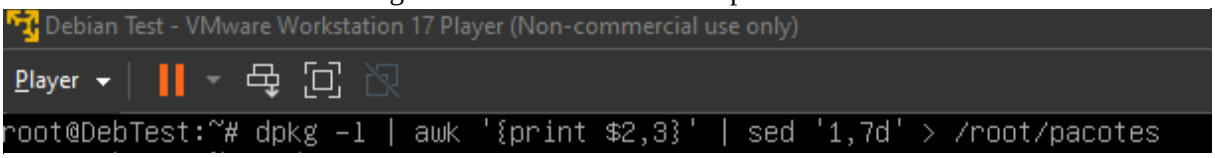
Uma das principais vantagens que os atacantes possuem ao tentar penetrar em sistemas reside no mal gerenciamento ou falta dele nas atualizações de pacotes de software e aplicativos. Muitas vezes, o software instalado está com versões antigas ou parâmetros originais, e a não manutenção adequada pode apresentar vulnerabilidades, que possa facilitar a vida dos atacantes no uso de códigos maliciosos, roubar informações ou realizar outras atividades que prejudique

o sistema, é crucial que os administradores mantenham seus sistemas regularmente atualizados com versões mais recente.

2.2 CONFIGURAÇÃO SEGURA

Ajustar as configurações do servidor web para eliminar ou reduzir vulnerabilidades conhecidas, como desabilitar serviços desnecessários que tenham funcionalidades não necessárias, configurar corretamente os parâmetros de segurança e limitar permissões, como a utilização de menor recurso e menor privilégio, é um ótimo princípio do conceito de *hardening*.

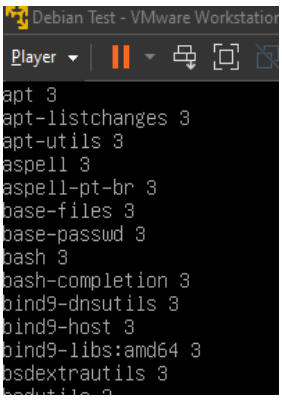
Figura 2 Verificando a lista de pacotes instalados



Fonte: Autoria Própria (2023).

A Figura 2 mostra um comado que verifica a lista de pacotes instalados e grava o resultado em um arquivo dentro do diretório */root/pacotes* do sistema operacional *Linux* distribuição conhecida como *Debian*.

Figura 3 Retorno da busca



Fonte: Autoria Própria (2023).

A Figura 3 mostra a lista dos pacotes instalados no sistema, resultado de busca do comando executado conforme Figura 2.

De acordo com Melo (2017, p.6), a *CIS Security* enfatiza a importância de desinstalar programas não utilizados, pois esses programas podem ser explorados por um *exploit*, que

consiste em um código desenvolvido para explorar vulnerabilidades em um programa específico, permitindo o acesso não autorizado ao sistema. Essa exploração pode resultar em ataques locais ou remotos, dependendo do programa em questão.

Quadro 1 Informações genéricas sobre a probabilidade e os impactos de ameaças

Agente de ameaça	**Explorabilidade**	**Prevalência de Fraqueza**	**Fraqueza Detectável**	**Impactos técnicos**	**Impactos nos negócios**
Aplicação específica	Fácil: 3	Generalizada: 3	Fácil: 3	Grave: 3	Empresas específicas
	Médio: 2	Comum: 2	Médio: 2	Moderado: 2	
	Difícil: 1	Pouco comum: 1	Difícil: 1	Pequeno: 1	

Fonte: OWASP 2017.

O quadro 1 auxilia a leitura e interpretação das avaliações de risco associadas as configurações incorretas de segurança, indicadas na Figura 4.

Figura 4 Configuração incorreta de segurança

Fonte: OWASP Top 10 2021

2.3 GERENCIAMENTO DE AUTENTICAÇÃO E CONTROLE DE ACESSO

Para aumentar a segurança ao acessar o servidor web, é recomendado implementar uma forma de autenticação robusta, como a autenticação de dois fatores, além disso, é essencial utilizar senhas fortes e evitar o uso de credenciais padrão que possam ser facilmente adivinhadas ou exploradas. Para um reforço da proteção é aconselhável limitar o acesso aos recursos sensíveis por meio da aplicação de restrições de acesso baseados em *funções Role Based Access Control* (Controle de Acesso Baseado em Funções) com base nessas funções permite a aplicação de princípio do menor privilégio, e configurar cuidadosamente as permissões de arquivos e diretórios. Para que dessa forma, apenas usuários autorizados terão acesso aos recursos críticos do servidor, reduzindo os riscos e diminuindo o acesso não autorizado ou eventual exploração de vulnerabilidades.

A implementação de políticas de senhas de maneira apropriada, ajuda com uma camada a mais para proteção, ainda mais quando senhas robustas são estabelecidas, como um parâmetro de comprimento mínimo, uma combinação de caractere, e impedindo que senhas antigas sejam reutilizadas, as configuração são feitos acessando o diretório */etc/pam.d/common-password* com o editor de sua preferência.

Inclusão da opção minlen=X para estabelecer um comprimento mínimo para a senha. Substitua o "X" pelo número desejado como mínimo. Por exemplo, para definir um comprimento mínimo de 8 caracteres, adicione a linha abaixo:

➔ *password requisite pam_cracklib.so retry=3 minlen=8*

Inclusão de caracteres especiais, acrescente a opção dcredit=X para definir a quantidade mínima de caracteres especiais necessários. Substitua o "X" pelo número desejado como mínimo. Por exemplo, para requerer pelo menos um caractere especial, adicione a seguinte linha:

➔ password requisite pam_cracklib.so retry=3 dcredit=1

Renovação do histórico de senhas, acrescente a opção remember=X para estipular a quantidade de senhas anteriores que devem ser lembradas. Substitua o "X" pelo número desejado. Por exemplo, para lembrar das últimas cinco senhas e evitar que os usuários as reutilizem, adicione a seguinte linha:

➔ *password sufficient pam_unix.so use_authtok sha512 shadow remember=5*

Figura 5 Política de segurança

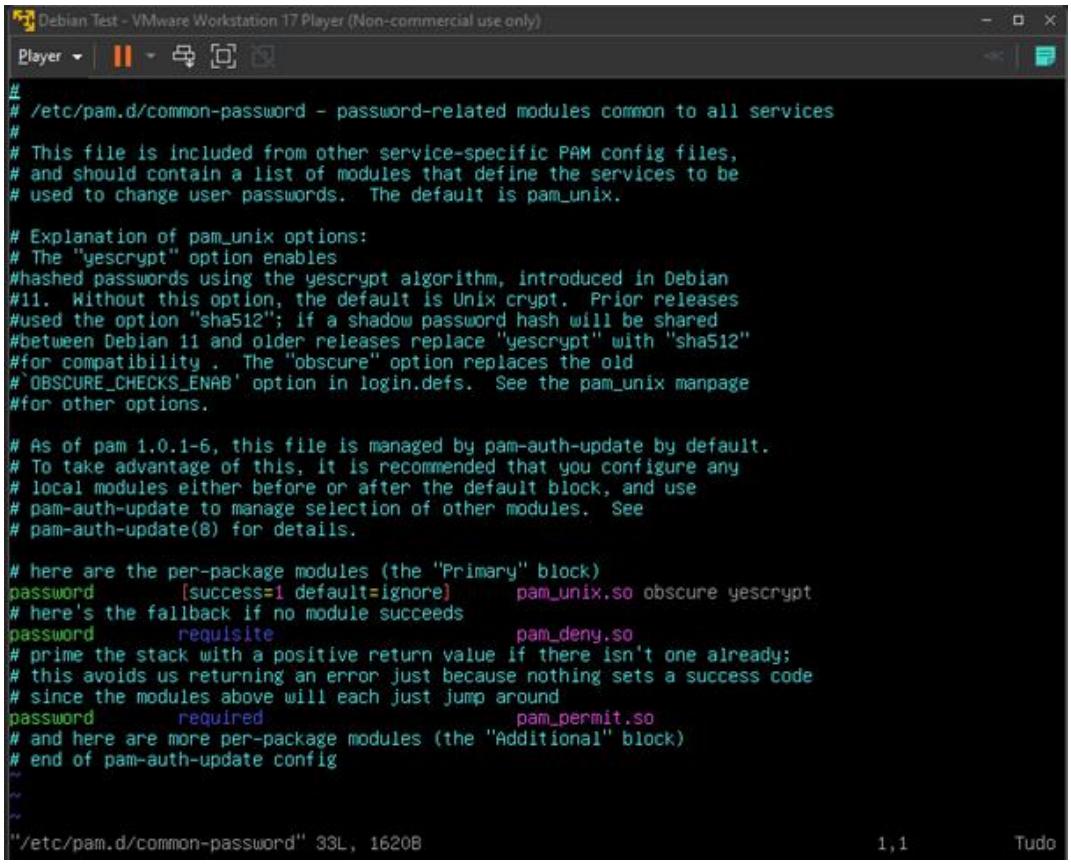
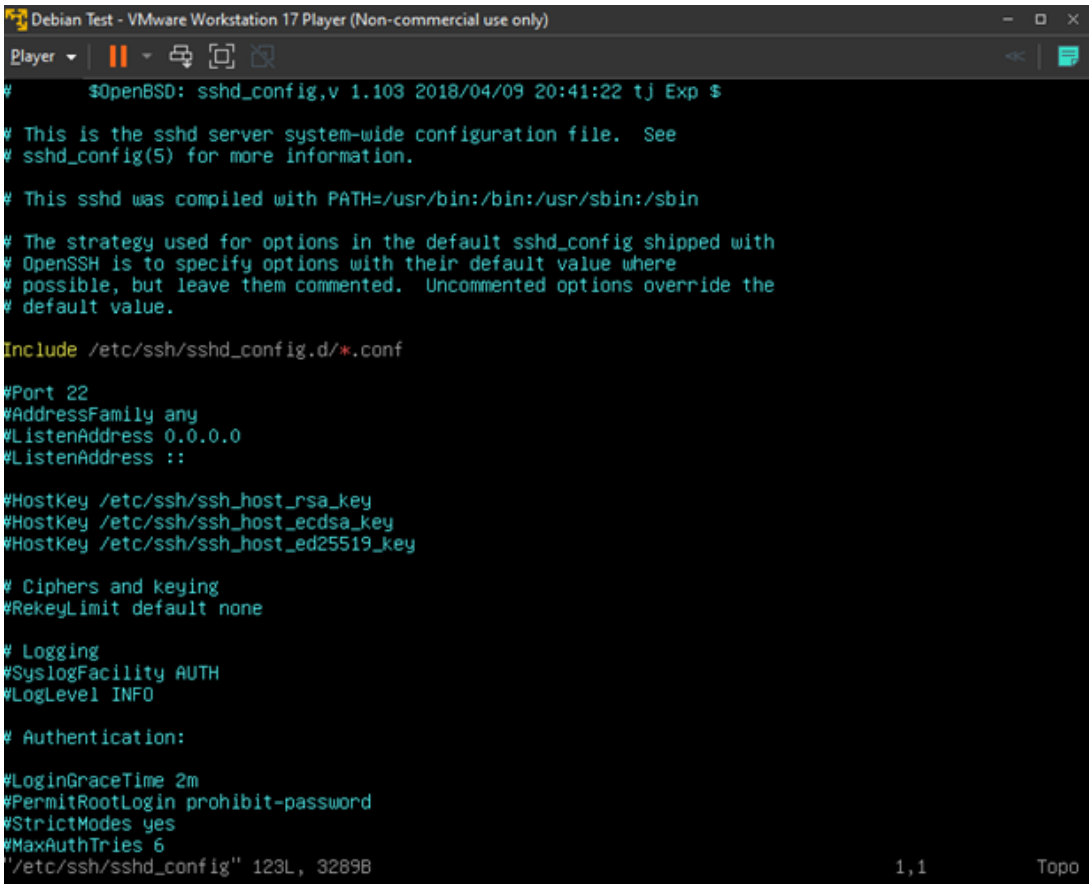


Figura 5 Política de segurança

Autenticação de múltiplos fatores, pensando na adoção de autenticação de múltiplos fatores, como autenticação de dois fatores (2FA) ou autenticação de fator duplo (MFA). Isso acrescenta uma camada adicional de segurança ao requerer uma segunda forma de autenticação além da senha.

Instale e configure uma solução e autenticação multifator, como *Google Authenticator* ou *FreeOTP*, edite o arquivo */etc/ssh/sshd_config* para habilitar autenticação multifatorial para conexões SSH. É possível configurar autenticação por chaves em vez de senhas, armazenar as chaves privadas com segurança e proteger com senhas fortes, Além de limitar o acesso às chaves SSH definindo as permissões corretas nos arquivos *authorized_keys*.

Figura 6 Configuração parâmetros do servidor

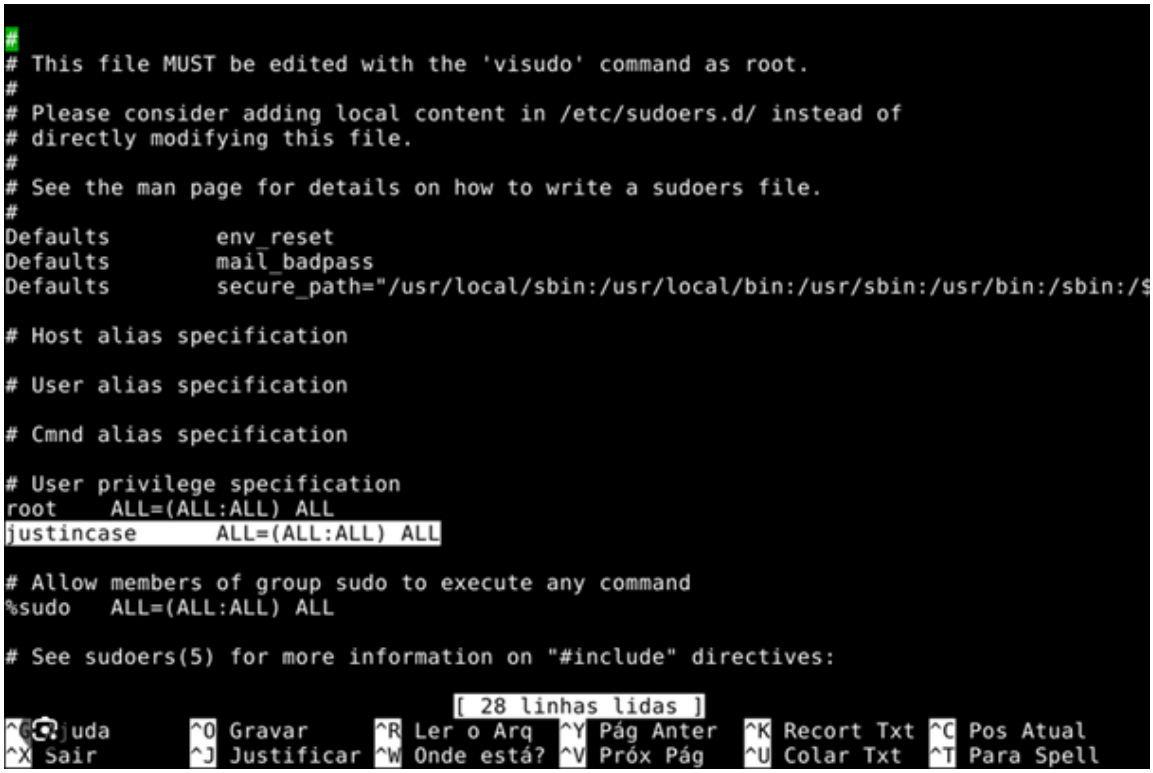


Fonte: Autoria Própria (2023).

Adotar o acesso seguro, com uma autenticação robusta, como autenticação de dois fatores, ao acessar o servidor web. Utilize senhas seguras e evitar o uso de credenciais padrão. Restringir o acesso a recursos sensíveis por meio da aplicação de restrições baseadas em funções (RBAC) e configurar corretamente as permissões de arquivos e diretórios.

Utilize o comando *sudo* para estabelecer e controlar as permissões de acesso dos usuários com base em suas funções. Configure as políticas de *sudo* no arquivo */etc/sudoers* para definir quais usuários têm acesso a quais comandos.

Figura 7 Acesso seguro



Fonte: Fonte: https://elias.praciano.com/2015/11/como-instalar-e-configurar-o-sudo-no-debian/

2.4 PROTEÇÃO CONTRA ATAQUES

"Sistemas de *firewalls* são importantes em um projeto de segurança; todavia, sozinhos não têm como garantir a segurança de uma rede de computadores. Demandam-se outros mecanismos, assim como uma administração proativa. No entanto, sua importância é notória, o que é abordado na norma internacional de segurança ISO/IEC 27002:2008. Essa norma recomenda a segregação da rede e a implementação de proteção dos serviços disponibilizados contra acessos não autorizados, destacando o papel dos sistemas de firewall na criação e gerenciamento de perímetros de redes adequadamente protegidos" (MELO, 2017, p.149).

2.5 CERTIFICADOS SSL/TLS

SL (*Secure Sockets Layer*) e TLS (*Transport Layer Security*) são protocolos de segurança usados para criar conexões seguras na Internet. Eles garantem a confidencialidade e a integridade dos dados transmitidos entre um cliente e um servidor, utilizando criptografia. A

criptografia é baseada em cifras, que são algoritmos matemáticos para codificar e decodificar os dados. O SSL/TLS usa tanto criptografia assimétrica (chave pública/privada) quanto criptografia simétrica (mesma chave) para proteger as informações durante a transmissão. *OpenSSL* é uma biblioteca de software de código aberto amplamente utilizada para implementar os protocolos SSL/TLS. Ela oferece funcionalidades criptográficas, como geração de chaves, seleção de cifras e autenticação, sendo comumente empregada em aplicativos e servidores da web para garantir a segurança das comunicações.

2.6 MONITORAMENTO E REGISTROS

De acordo com Turnbull (2005, p.233), manter um ambiente seguro requer o monitoramento adequado por meio de logs, que são registros detalhados das atividades em sistemas e aplicativos. Embora muitos sistemas e aplicativos possuam opções de log padrão, ao lidar com segurança, é necessário investigar mais a fundo os logs para obter uma compreensão mais completa do cenário. Os logs são valiosos tanto para a segurança quanto para invasores, pois podem conter informações cruciais sobre os sistemas e sua segurança, sendo frequentemente visados por invasores em busca de informações. Para proteger adequadamente os logs, é necessário garantir que os arquivos de log e o diretório onde estão armazenados estejam protegidos contra acesso não autorizado. Além disso, caso os logs sejam transmitidos pela rede para um servidor centralizado, é importante assegurar que não sejam interceptados ou desviados por terceiros.

3 PROCEDIMENTOS METODOLÓGICOS

A segurança dos servidores *web* desempenha um papel crucial na era digital, especialmente quando se trata de proteger informações sensíveis. Uma violação de segurança em servidores *web* pode ter consequências graves, como a perda de dados, interrupção dos serviços e danos irreparáveis à reputação de uma organização. Para evitar esses cenários indesejados, é essencial adotar medidas proativas, como o *hardening*, para fortalecer a segurança dos servidores web. No contexto deste artigo, foi utilizado o método de *scanner* de vulnerabilidades para identificar e mitigar possíveis pontos de falha que podem ser explorados por um atacante. Esse método permite uma avaliação abrangente da segurança do servidor *web*, identificando vulnerabilidades e fornecendo *insights* sobre como corrigi-las. Dessa forma, as

organizações podem tomar medidas proativas para fortalecer suas defesas e garantir a segurança de suas informações sensíveis.

O servidor web utilizado possui a seguinte configuração:

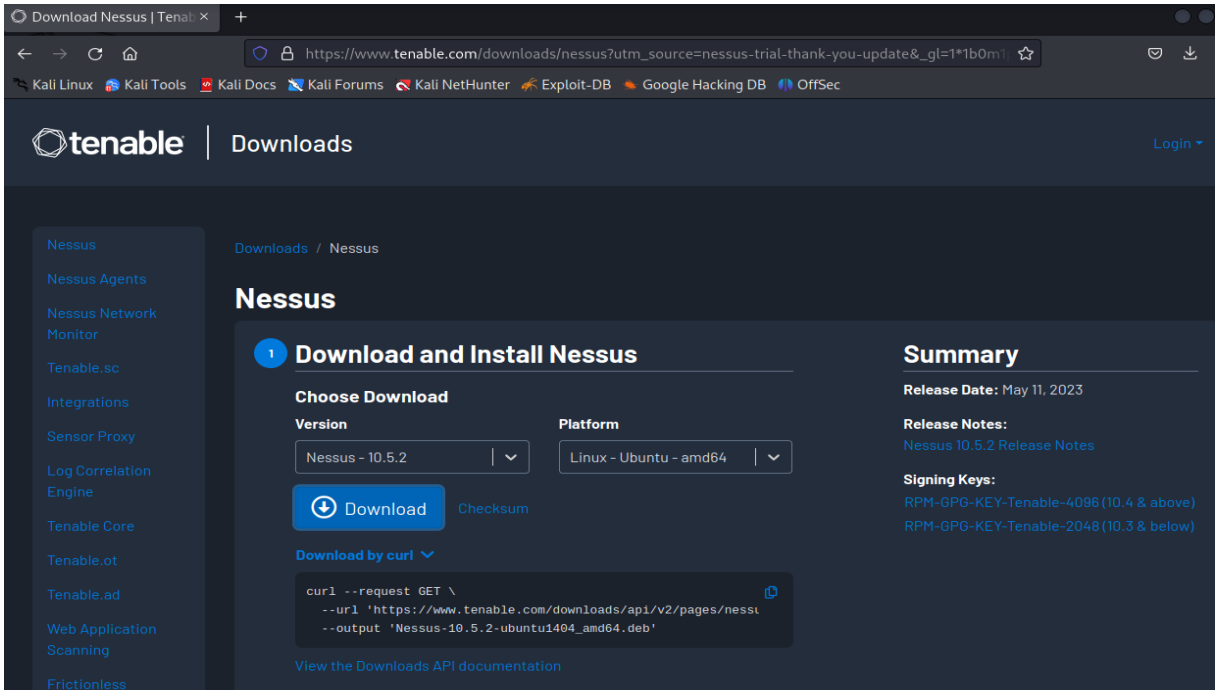
- Linux Debian 10
- Apache 2.4.38
- PHP 7.3.14-1
- PostgreSQL 11.7

A ferramenta de scanner utilizada durante a pesquisa foi o *Tenable Nessus* 10.5.2, que possui um banco de dados atualizado com as vulnerabilidades conhecidas e mais de 152 plugins de produtos e *appliances* diferentes.

3.1 Instalação do Nessus

Acesse a página de download https://www.tenable.com/downloads/nessus para realizar verificar a URL do pacote da versão corrente do *Nessus*.

Figura 8 Página de download do Nessus

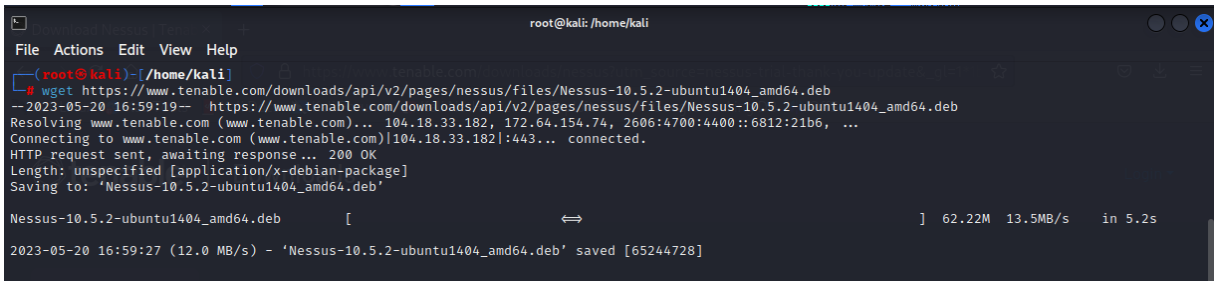


Fonte: Autoria Própria (2023).

Uma vez checada a URL que será utilizada, use o comando *wget* para realizar o download do pacote:

wget *https://www.tenable.com/downloads/api/v2/pages/nessus/files/Nessus-10.5.2-debian10_amd64.deb*

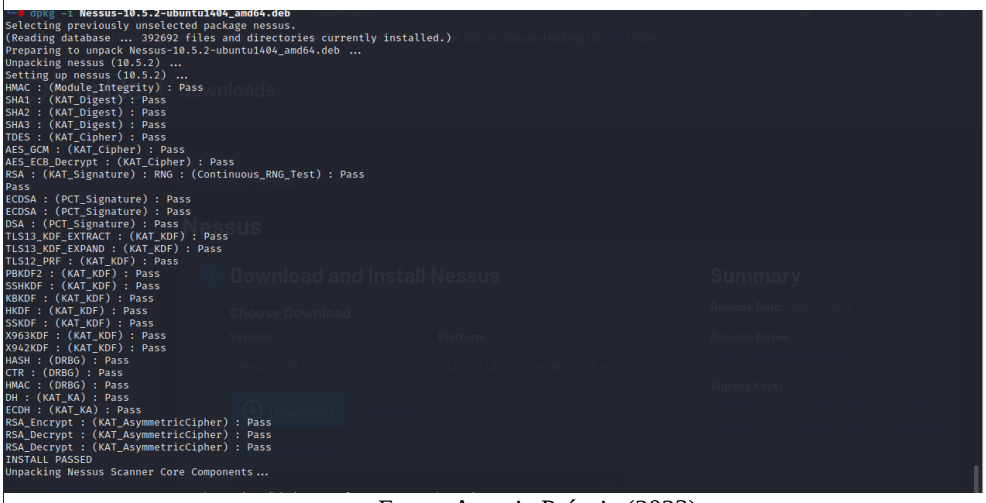
Figura 9 Download do pacote de instalação



Fonte: Autoria Própria (2023).

Utilize o comando *dpkg -i Nessus-10.5.2-debian10_amd64.deb* para iniciar a instalação:

Figura 10 Iniciando a instalação do Nessus no Linux



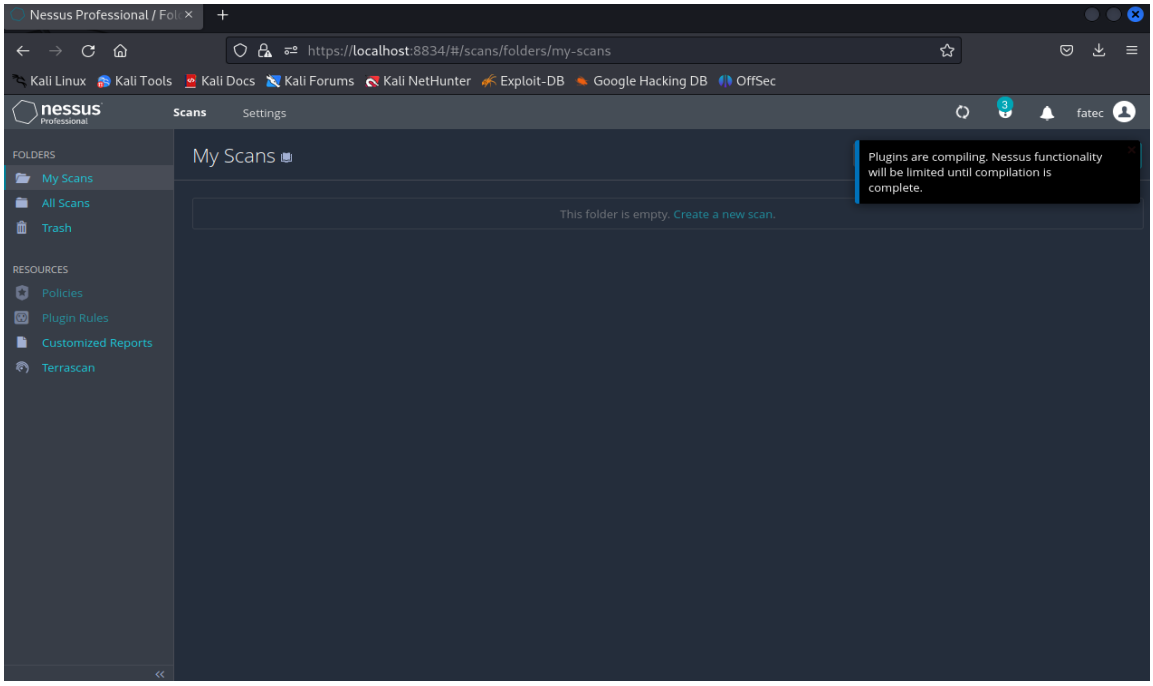
Fonte: Autoria Própria (2023).

3.2 CONFIGURAÇÃO DO NESSUS

Depois de instalado, devemos acessar a interface web para realizar as configurações do software antes da primeira utilização. O endereço padrão é https://localhost:8834.

Após a instalação ser finalizada, será exibida a tela principal da ferramenta onde será informado que os plugins estão sendo compilados, aguarde até que seja finalizado:

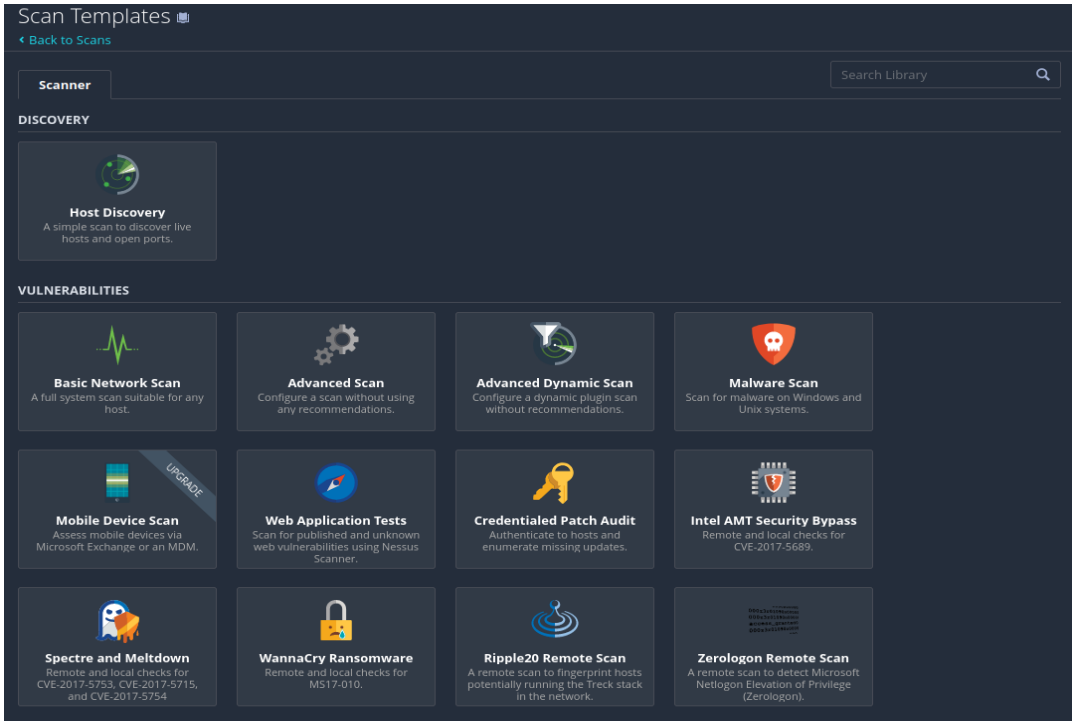
Figura 11 Plugins a serem compilados



Fonte: Autoria Própria (2023).

Clique em *create* a new *scan* para informar os dados do servidor a ser escaneado, selecione o tipo de *scan* que será realizado, existe a opção de utilizar *templates* específicos para vulnerabilidades conhecidas. Como queremos encontrar qualquer vulnerabilidade existente iremos escolher o *template Advanced Scanner* que não possui nenhuma especificidade na busca:

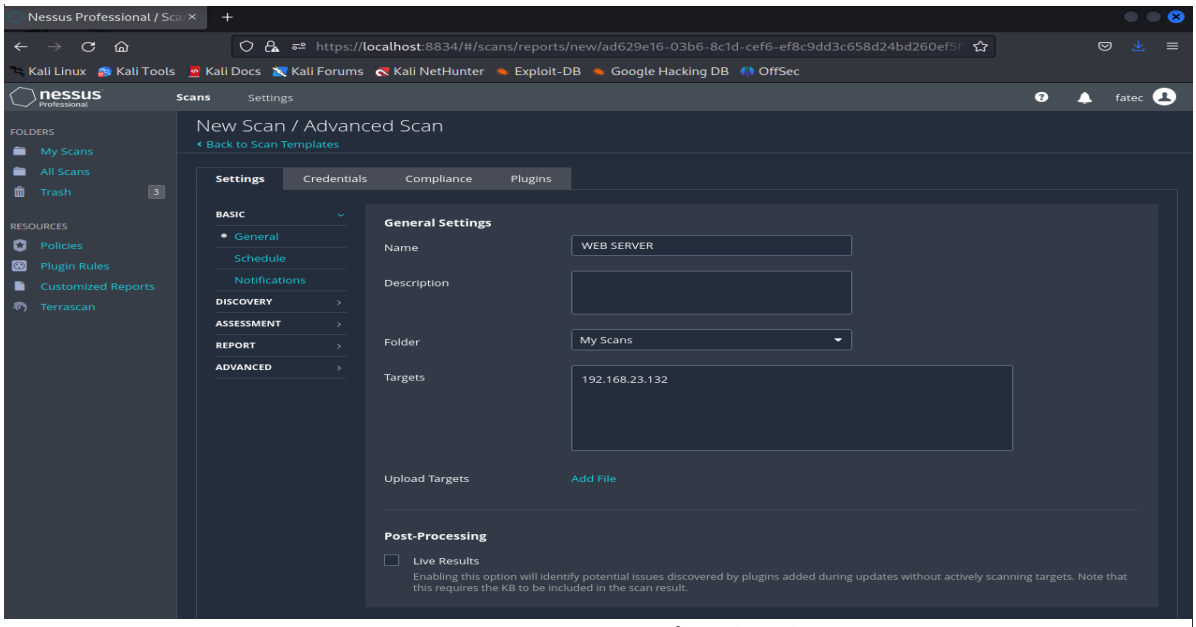
Figura 12 Scan Templates



Fonte: Autoria Própria (2023).

Insira o nome, a descrição e endereço do host a ser escaneado. Pode ser definido um host individual ou uma rede completa para ser verificada, clique em *save*:

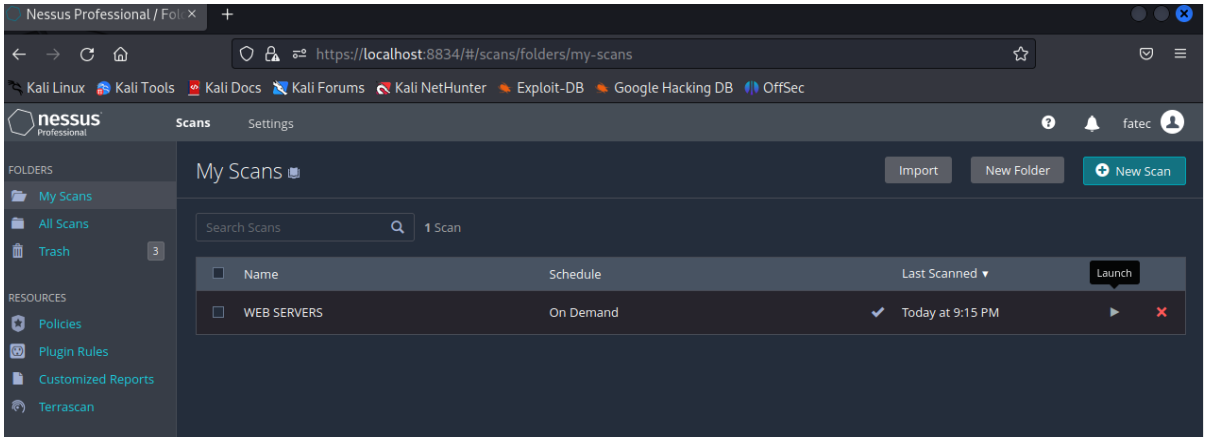
Figura 13 Configurando os parâmetros do Scan



Fonte: Autoria Própria (2023).

Volte para a pasta onde o *scan* foi salvo, e clique em *launch*, o *scan* será iniciado:

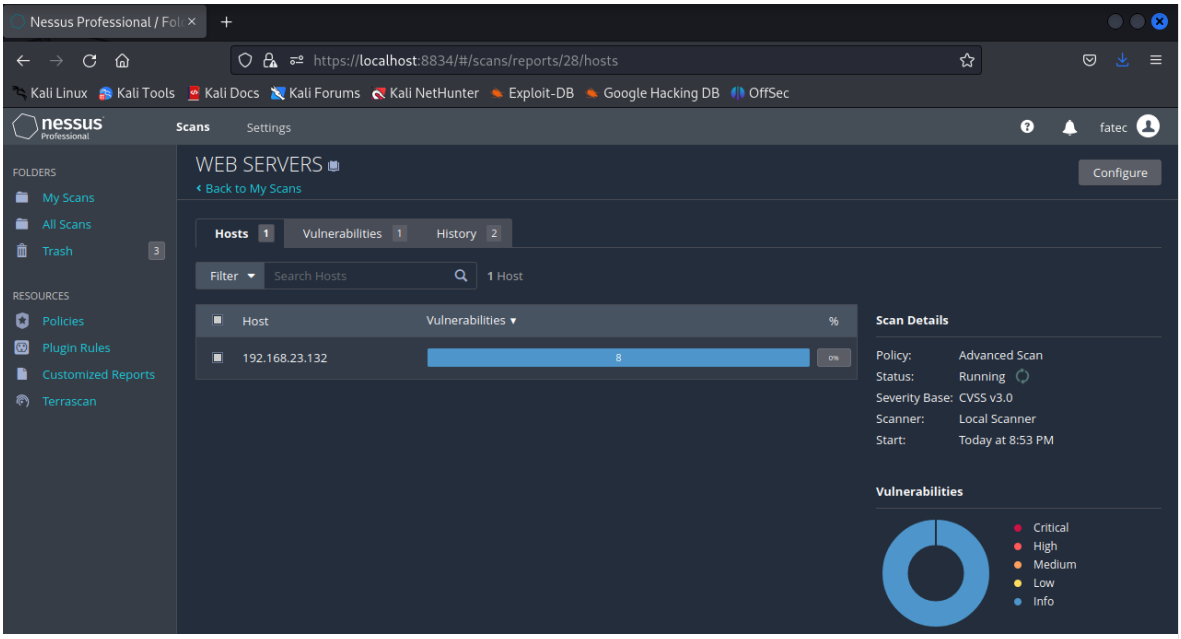
Figura 14 Lista de Scans configurados



Fonte: Autoria Própria (2023).

Será iniciado o *scan* e populado em tempo real com as quantidades de vulnerabilidades encontradas e a descrição delas:

Figura 15 Scan iniciado

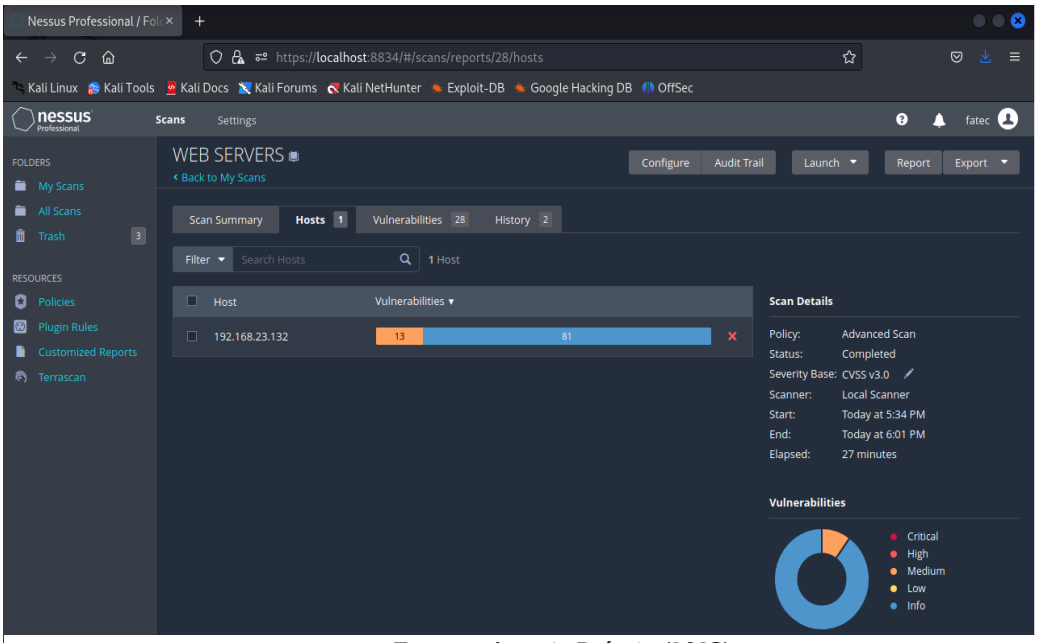


Fonte: Autoria Própria (2023).

4 RESULTADOS E DISCUSSÃO

Com o escaneamento finalizado, podemos observar as vulnerabilidades encontradas, e com isso encontrar a forma assertiva de corrigi-las. O *Nessus* fornece os resultados em tela ou em forma de relatório com detalhes e classificação conforme ilustrados na Figura a seguir:

Figura 16 Resultado do Scan



Fonte: – Autoria Própria (2023).

Podemos observar que nos resultados aparecem algumas vulnerabilidades listadas no TOP 10 da *OWASP*, mostrando que a negligência e a falta de atenção na implantação de recursos para hospedagem de aplicações web pode facilmente esconder perigos que podem trazer impactos negativos, sejam eles vazamento de dados sensíveis ou indisponibilidade completa dos serviços.

Com a riqueza de informação que o *Nessus* traz, podemos agir de forma proativa, mantendo os escaneamentos frequentes com execuções agendadas e com isso deixar o *hardening* do ambiente em dia logo que as vulnerabilidades forem descobertas.

5 CONSIDERAÇÕES FINAIS

O *hardening* de servidores web desempenha um papel fundamental na proteção dos dados e na garantia da segurança dos serviços online. Neste artigo, exploramos os conceitos, as técnicas e as melhores práticas relacionadas ao *hardening*, abordando aspectos como configuração segura, criptografia, gerenciamento de autenticação e controle de acesso, mitigação de ataques comuns e utilização de ferramentas especializadas.

Ao longo da discussão, destacamos a importância de implementar uma abordagem em camadas, combinando várias técnicas de *hardening* para fortalecer a segurança dos servidores web. É crucial reconhecer que o *hardening* não é uma solução estática, mas um processo

contínuo que requer monitoramento constante, atualizações e adaptações para enfrentar ameaças emergentes e com as ferramentas de apoio, como o *Nessus*, podemos ter uma visão mais abrangente para facilitar essa tarefa.

Em conclusão, implementar medidas de *hardening* adequadas, combinadas com uma abordagem em camadas, contribui para fortalecer a segurança dos servidores web, proteger dados valiosos e manter a confiança dos usuários. Ao adotar as práticas e recomendações discutidas neste artigo, as organizações podem estar mais bem preparadas para enfrentar os desafios da segurança cibernética e garantir a integridade e disponibilidade de seus serviços online.

REFERÊNCIAS

MELO, Sandro. *Hardening* em Linux. Rio de Janeiro: Editora Rede Nacional de Ensino e Pesquisa – RNP, 2014.

OWASP. OWASP Top 10 - 2017: The Ten Most Critical Web Application Security Risks. 2017. PDF. Disponível em: https://wiki.owasp.org/images/0/06/OWASP_Top_10-2017-pt_pt.pdf Acesso em:15/06/2023.

TURNBULL, James. Hardening Linux. United States: Editora Apress, 2005.

TENABLE. Tenable - Nessus. Disponível em: https://pt-br.tenable.com/products/nessus Acesso em: 14/06/2023.