Fatecnológica Jahu

ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

FORTALECENDO A SEGURANÇA DE SERVIDORES WEB – UM ESTUDO SOBRE HARDENING

STRENGTHENING THE SECURITY OF WEB SERVERS – A STUDY IN HARDENING

Evandro Ferreira Melo Pires
Faculdade de Tecnologia de Araraquara, evandro.pires@fatec.sp.gov.br

Leonardo Araújo Santos
Faculdade de Tecnologia de Araraquara, leonardo.santos229@fatec.sp.gov.br

João Emmanuel D' Alkmin Neves
Faculdade de Tecnologia de Araraquara, joao.neves11@fatec.sp.gov.br

DOI: https://doi.org/10.54628/issn2763-5600.v19.1.2025.268

RESUMO

Este artigo aborda a importância do *hardening* de servidores *web* como estratégia essencial para fortalecer a segurança dos sistemas. O *hardening* consiste na adoção de práticas e configurações que visam minimizar vulnerabilidades e prevenir ataques. A atualização contínua dos *softwares* dos servidores é destacada, uma vez que versões desatualizadas podem expor os sistemas a riscos de acesso não autorizado. O artigo enfatiza a necessidade de configurar corretamente os servidores, desativando serviços desnecessários e definindo permissões apropriadas. A implementação de certificados *SSL/TLS* é sugerida para assegurar a confidencialidade e integridade das transmissões de dados. Além disso, a utilização de autenticação robusta, senhas fortes e controle de acesso baseado em funções é considerada crucial para proteger recursos sensíveis. A presença de ferramentas de segurança, como *firewalls*, e a definição de políticas de senhas adequadas são igualmente destacadas. O monitoramento contínuo e a proteção de *logs* de atividades também são abordados, pois os registros podem ser alvo de ataques. A ferramenta *Nessus* é mencionada como um recurso eficaz para análise de vulnerabilidades e conformidade, sendo utilizada para identificar falhas e fornece recomendações de correção.

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

ABSTRACT

This scientific paper discusses the importance of web server hardening as a critical strategy to enhance system security. Hardening involves adopting practices and configurations aimed at minimizing vulnerabilities and preventing attacks. Continuous software updates are emphasized, as outdated versions can expose systems to unauthorized access risks. The paper highlights the need for proper server configuration by disabling unnecessary services and defining appropriate permissions. The implementation of SSL/TLS certificates is recommended to ensure the confidentiality and integrity of data transmissions. Additionally, the use of robust authentication, strong passwords, and role-based access control is considered essential for protecting sensitive resources. The deployment of security tools, such as firewalls, and the establishment of proper password policies are also emphasized. Continuous monitoring and protection of activity logs are discussed, as logs can be targeted by attackers. The Nessus tool is presented as an effective resource for vulnerability analysis and compliance checks, being used to identify weaknesses and provide corrective recommendations.

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

ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

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 (Neves, 2024).

O problema central reside na constante exposição dos servidores a ataques cibernéticos, que exploram falhas de configuração, vulnerabilidades de *software* ou falta de práticas de segurança adequadas, comprometendo a confidencialidade, a disponibilidade e a integridade dos dados.

A hipótese que guia este estudo é que a implementação de práticas de *hardening* em servidores web pode reduzir significativamente as vulnerabilidades e mitigar os riscos de ataques cibernéticos.

Conforme abordado por Turnbull (2005), o termo em inglês *hardening*, traduzido literalmente como endurecimento, refere-se ao processo de fortalecimento de um sistema operacional por meio da implementação de técnicas específicas de configuração e controle, com o intuito de aprimorar sua segurança. No contexto da segurança da informação, esse conceito envolve a mitigação de vulnerabilidades, visando reduzir os riscos de exploração e, assim, proteger o sistema contra possíveis ameaças e ataques.

O objetivo deste artigo é analisar as técnicas e estratégias de *hardening* aplicáveis a servidores web, com foco na configuração segura, no gerenciamento de autenticação, no controle de acesso e nas práticas para mitigar ataques comuns, como injeção de código, *Cross-Site Scripting* e ataques de negação de serviço. Além disso, será explorada a utilização da ferramenta *Nessus* para a análise de vulnerabilidades e a geração de relatórios detalhados, contribuindo para um monitoramento contínuo da segurança.

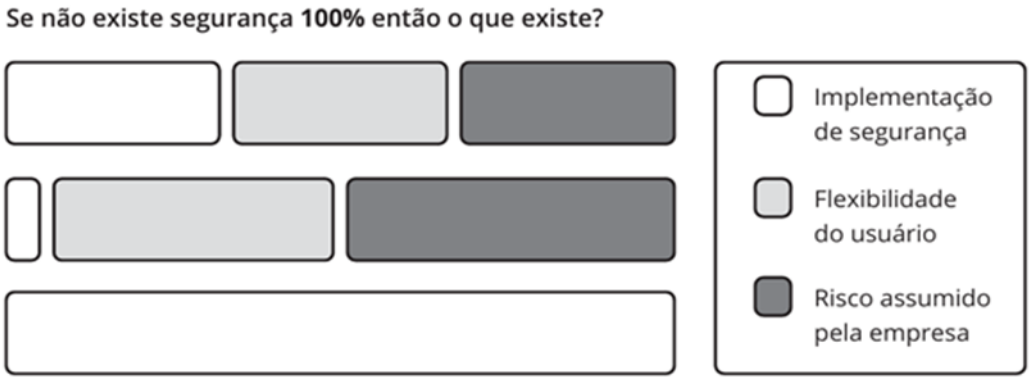
A justificativa para esta pesquisa está no crescente número de ataques direcionados a servidores web, cuja segurança inadequada pode resultar em danos financeiros e reputacionais para as organizações. Portanto, adotar práticas eficazes de *hardening* e utilizar ferramentas de análise de vulnerabilidades é fundamental para proteger dados sensíveis e garantir a continuidade dos serviços online, um componente essencial para a confiança dos usuários na Internet (Santos; Neves, 2023).

Fatecnológica Jahu
ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

2 FUNDAMENTAÇÃO TEÓRICA

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.

Figura 1 - Segurança, risco, flexibilidade



Fonte: Melo (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 (Souza *et al.*, 2024). Portanto 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 (Moura; D'Alkmin Neves, 2021). 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

ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

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

2.2 CONFIGURAÇÃO SEGURA

A configuração segura de sistemas é crucial para assegurar a proteção das informações e prevenir possíveis ameaças. É importante adotar o princípio do menor privilégio, garantindo que usuários e processos tenham acesso apenas ao necessário para o funcionamento do sistema. A implementação de *firewalls*, como o *iptables*, é essencial para controlar o tráfego de rede, enquanto o uso de autenticação robusta, com senhas fortes e autenticação multifatorial, é altamente recomendada (Neves *et al.*, 2023). Além disso, serviços e módulos desnecessários devem ser desabilitados, e ferramentas de auditoria e monitoramento, como o *auditd*, devem ser configuradas. A criptografia de dados, tanto em repouso quanto em trânsito, e a escolha de sistemas de arquivos seguros, como *ext4* ou *xfs*, são medidas importantes para proteger os dados contra acessos não autorizados.

O Quadro 1 apresenta o comando utilizado para verificar a lista de pacotes instalados em um sistema operacional Linux, mais especificamente em uma distribuição Debian. Esse comando é fundamental para obter informações detalhadas sobre os pacotes que estão atualmente instalados, permitindo uma análise precisa do ambiente. Após a execução, o resultado da verificação é registrado em um arquivo, que é salvo no diretório /root/pacotes. Esse diretório, localizado na raiz do sistema, é específico do usuário administrador e serve como um local seguro para armazenar arquivos do sistema. A criação desse arquivo facilita o acompanhamento e a documentação dos pacotes instalados, o que pode ser útil para auditorias de segurança, gerenciamento de software e diagnóstico de possíveis falhas. Dessa forma, o comando não apenas coleta as informações necessárias, mas também organiza os dados de forma a permitir uma consulta eficiente no futuro, contribuindo para a manutenção da integridade e da gestão do sistema operacional.

Quadro 1 - Verificando a lista de pacotes instalados

Comando	`dpkg -l | awk '{print $2, $3}' | sed '1,7d' > /root/pacotes`

Fonte: Autoria Própria (2023)

O Quadro 2 lista os pacotes instalados no sistema, resultado de busca do comando executado conforme Quadro 1.

ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

Quadro 2 - Lista pacotes instalados

Pacote	Versão	Pacote	Versão
alsa-ucm-conf	1.2.8-1	amd64-microcode	3.20240820.1~deb12u1
anacron	2.3-36	apache2	2.4.62-1~deb12u2
apache2-bin	2.4.62-1~deb12u2	apache2-data	2.4.62-1~deb12u2
apache2-utils	2.4.62-1~deb12u2	apparmor	3.0.8-3
apt	2.6.1	apt-listchanges	3.24
apt-utils	2.6.1	aspell	0.60.8-4+b1
aspell-pt-br	20131030-18	avahi-autoipd	0.8-10
base-files	12.4+deb12u8	base-passwd	3.6.1
bash	5.2.15-2+b7	bash-completion	1:2.11-6
bind9-dnsutils	1:9.18.28-1~deb12u2	bind9-host	1:9.18.28-1~deb12u2
bind9-libs	1:9.18.28-1~deb12u2	bluetooth	5.66-1+deb12u2
bluez	5.66-1+deb12u2	bsdextrautils	2.38.1-5+deb12u2
bzip2	1.0.8-5+b1	ca-certificates	20230311
console-setup	1.221	console-setup-linux	1.221
coreutils	9.1-1	cpio	2.13+dfsg-7.1
cron	3.0pl1-162	cron-daemon-common	3.0pl1-162
dash	0.5.12-2	dbus	1.14.10-1~deb12u1
dbus-bin	1.14.10-1~deb12u1	dbus-daemon	1.14.10-1~deb12u1
dbus-session-bus-common	1.14.10-1~deb12u1	dbus-system-bus-common	1.14.10-1~deb12u1
dbus-user-session	1.14.10-1~deb12u1	debconf	1.5.82
debconf-i18n	1.5.82	debian-archive-keyring	2023.3+deb12u1
debian-faq	11.1	debianutils	5.7-0.5~deb12u1

Fonte: Autoria Própria (2023)

Segundo Melo (2017), a CIS Security destaca a importância de desinstalar programas não utilizados, pois esses softwares podem ser explorados por exploits, códigos desenvolvidos para aproveitar vulnerabilidades e permitir o acesso não autorizado ao sistema, o que pode levar a ataques locais ou remotos. O Quadro 3 facilita a leitura e interpretação das avaliações de risco relacionadas a configurações de segurança inadequadas.

Fatecnológica Jahu
ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

Quadro 3 - Informações genéricas sobre a probabilidade e os impactos de ameaças

Agente de ameaça	Explorabilidade	Prevalência de Fraqueza	Fraqueza Detectável	Impactos Técnicos	Impactos 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)

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* 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.

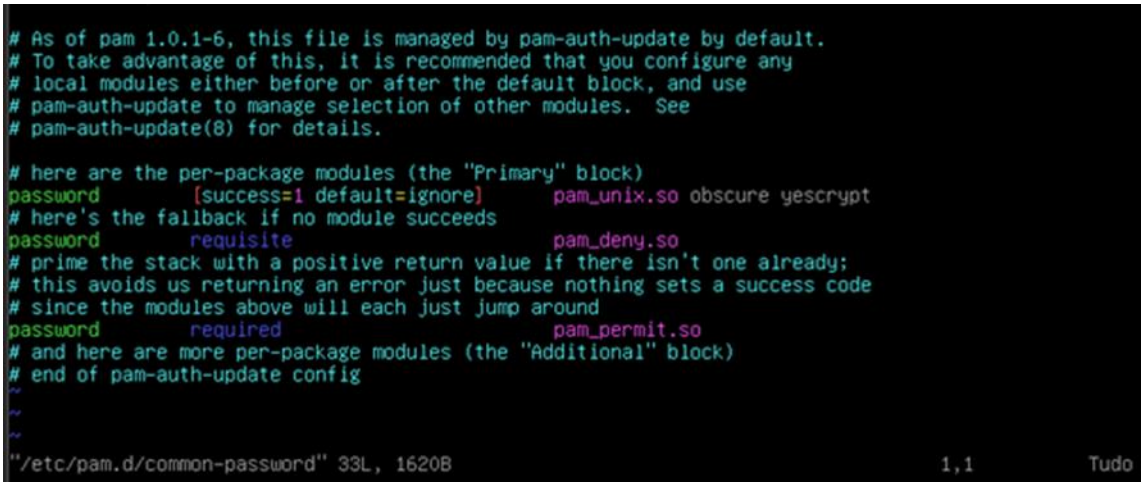
A implementação adequada de políticas de senhas oferece uma camada adicional de proteção, especialmente quando são estabelecidos requisitos como: comprimento mínimo, combinação de caracteres e a proibição de reutilização de senhas antigas. Essas configurações são realizadas acessando o diretório /etc./pam.d/common-password, conforme mostrado na Figura 3. A adoção de senhas robustas, aliada a essas políticas, fortalece significativamente a segurança do sistema.

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

A 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***

Fatecnológica
Jahu
ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

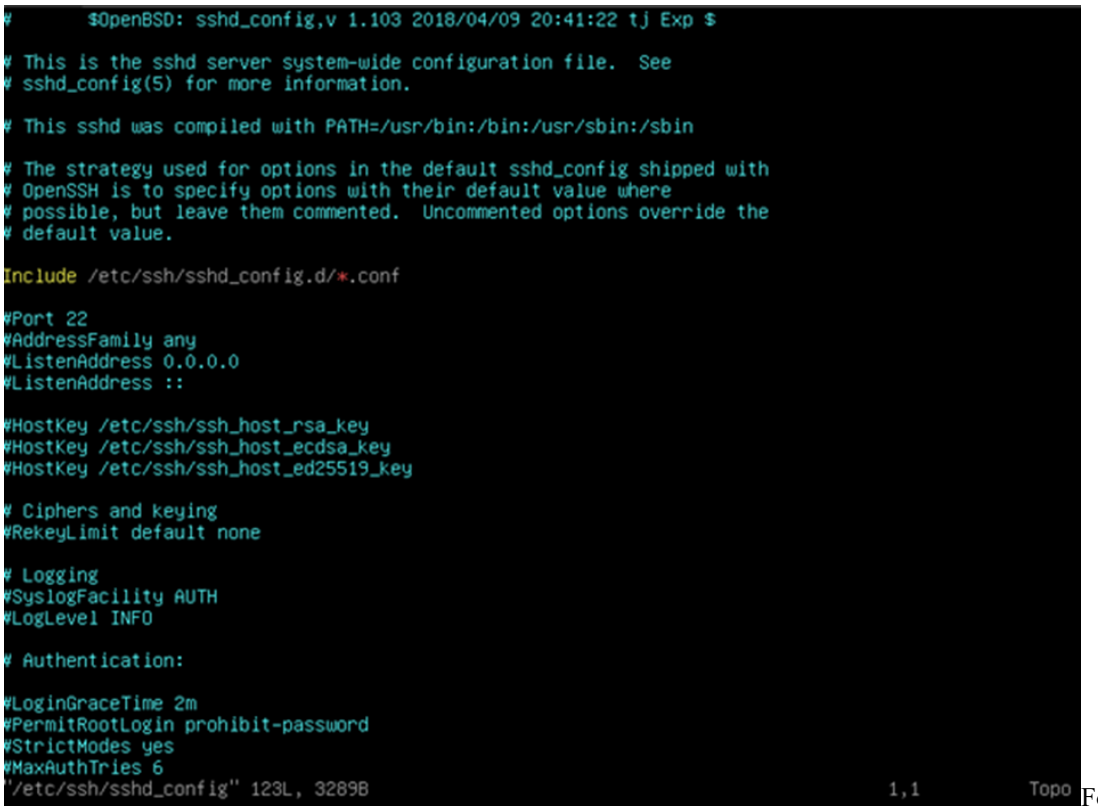
Figura 2 - Política de Segurança



Fone: Autoria Própria (2023)

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: ***password sufficient pam_unix.so use_authtok sha512 shadow remember=5.*** Conforme visto na Figura 4.

Figura 3 - Configuração parâmetros do servidor



Fonte: Autoria Própria (2023).

ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

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 mulifator 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.*

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.

Segundo Praciano (2023), o comando *sudo* deve ser utilizado para estabelecer e controlar as permissões de acesso dos usuários com base em suas funções. As políticas de *sudo* podem ser configuradas no arquivo /etc/sudoers, permitindo a definição precisa de quais usuários têm permissão para executar determinados comandos, conforme ilustrado na Figura 5.

Figura 4 - Acesso Seguro

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/s

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
justincase      ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:
```

Fonte: Praciano (2023)

ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

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 ISSO/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).

O *SSL* (*Secure Sockets Layer*) e o *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.

2.5 MONITORAMENTO E REGISTROS

De acordo com Turnbull (2005), 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.

ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

3 MATERIAIS E MÉTODOS

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 conhecidas 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 configuração:

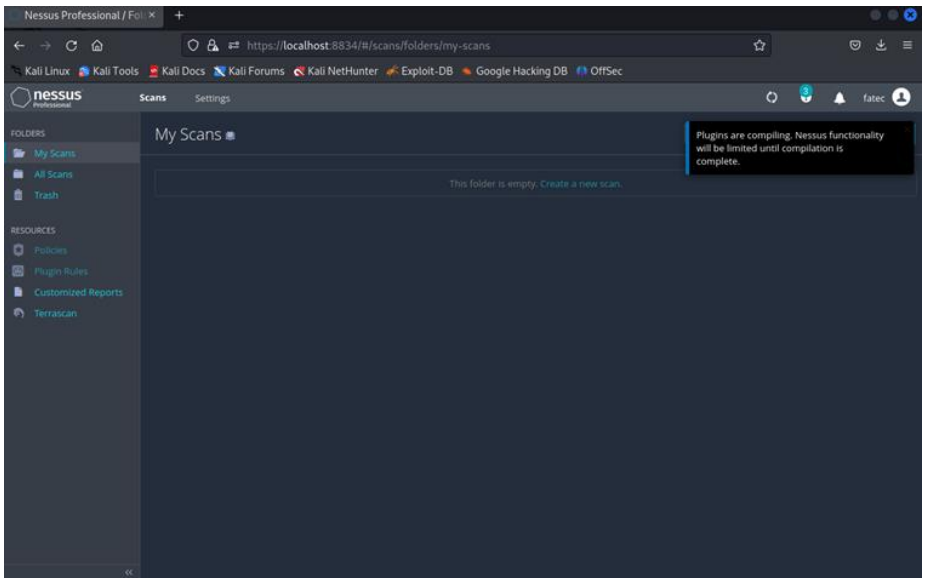
- Linux Debian 10 , Apache 2.4.38, PHP 7.3.14-1, PostgreSQL 11.7

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 CONFIGURAÇÃO DO NESSUS

Depois de instalado, acessar a interface *web* do 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, conforme mostrado na Fig. 6.

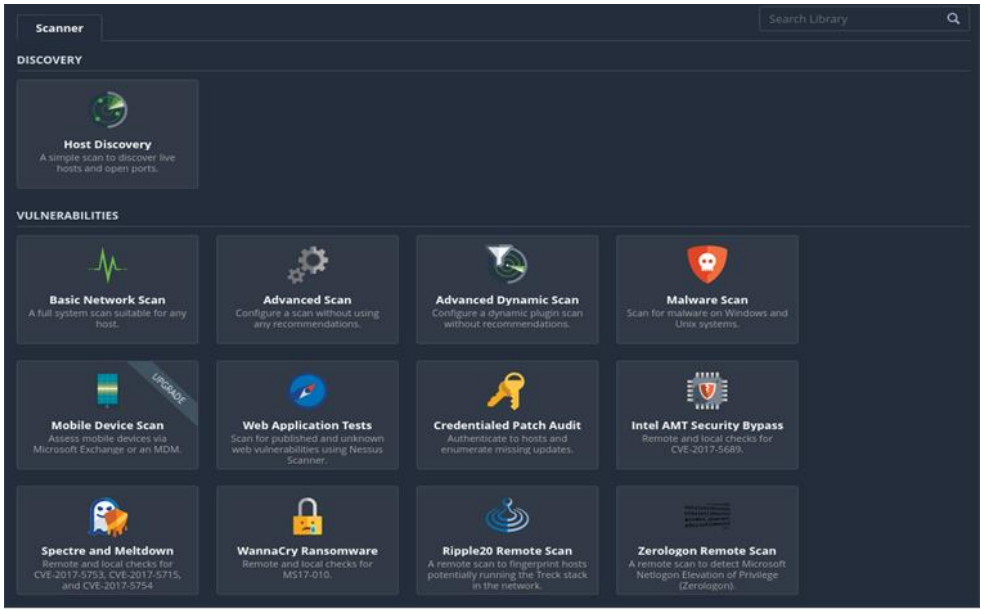
Figura 5 - *Plugins* a serem compilados



ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

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. Com o objetivo de encontrar qualquer vulnerabilidade existente será escolhido o *template Advanced Scanner* que não possui nenhuma especificidade na busca, conforme mostrado na Figura 7.

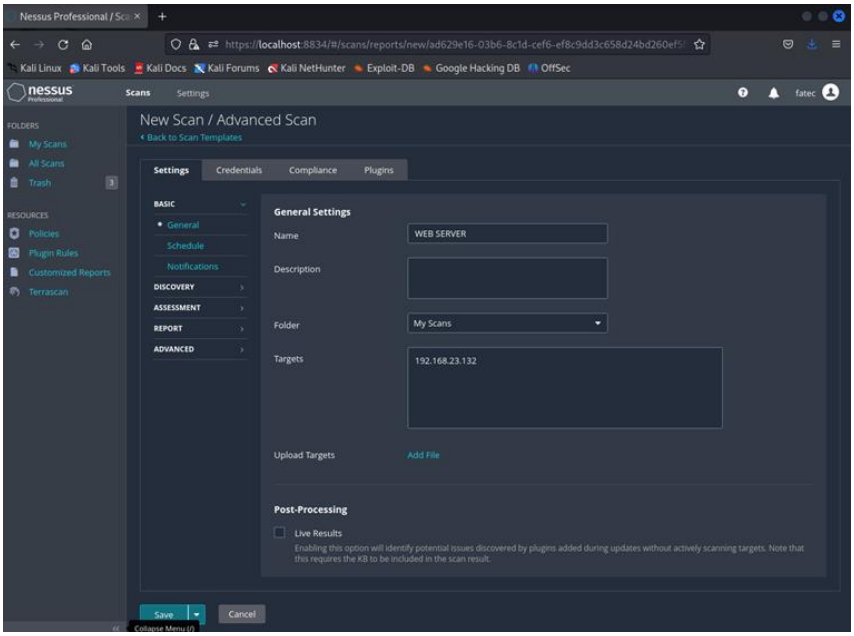
Figura 6 - *Scan Templates*



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 8).

Figura 7 - Configurando os parâmetros do *Scan*

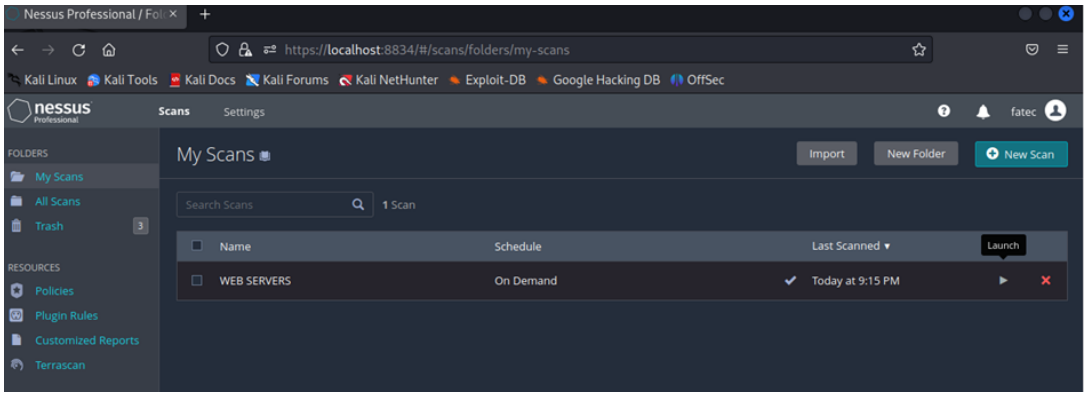


Fonte: Autoria Própria (2023)

Fatecnológica
Jahu
ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

Volte para a pasta onde o *scan* foi salvo, e clique em *launch*, o *scan* será iniciado, conforme mostrado na Figura 9.

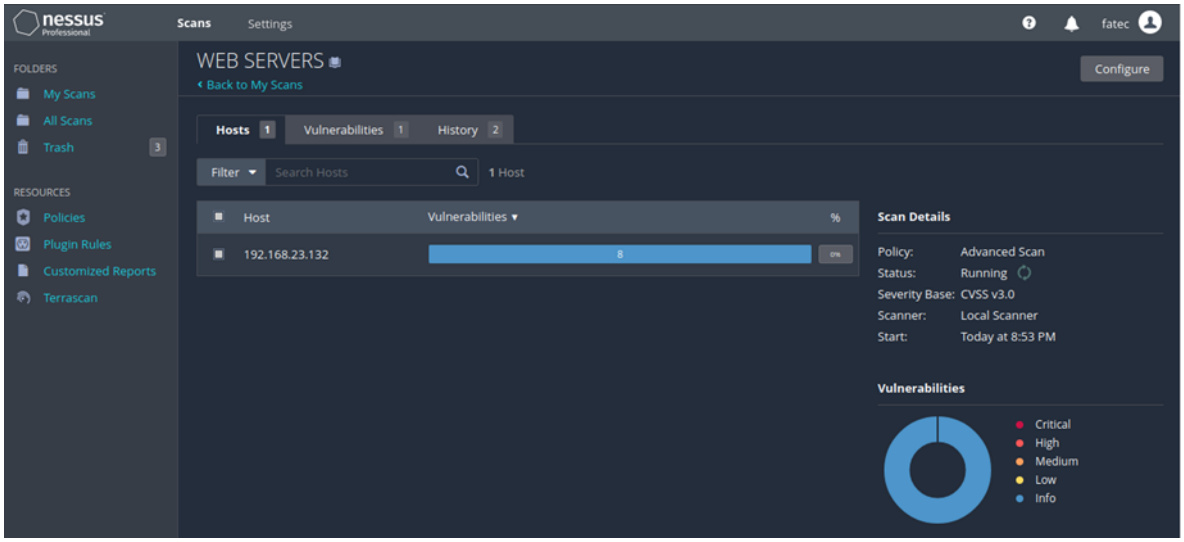
Figura 8 - Lista de *Scans* configurados



Fonte: Autoria Própria (2023)

Será iniciado o *scan* e populado em tempo real com as quantidades de vulnerabilidades encontradas e a descrição delas, conforme demonstrado na Figura 10.

Figura 9 - *Scan* iniciado



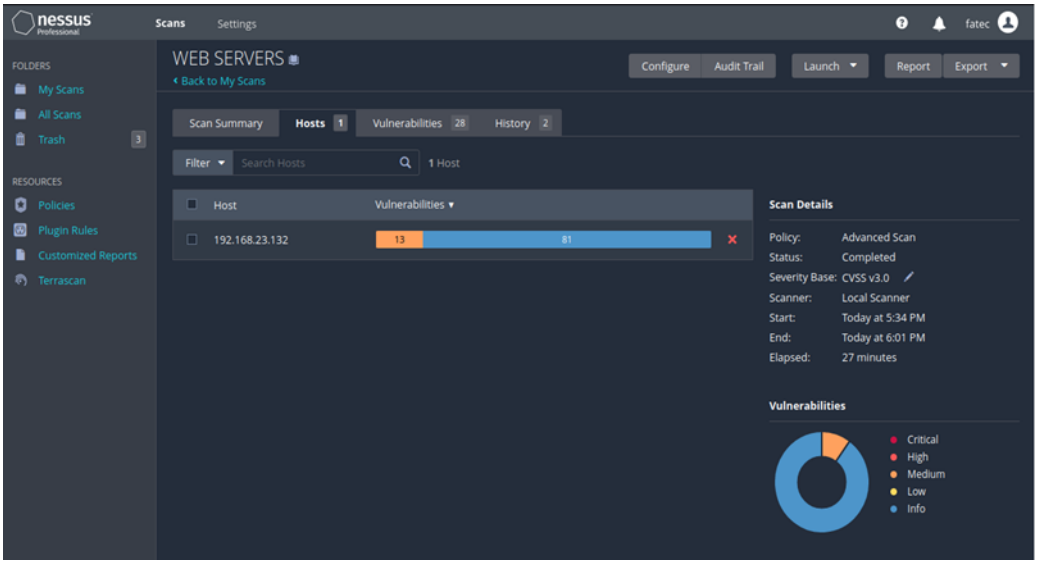
Fonte: Autoria Própria (2023)

4 RESULTADOS E DISCUSSÃO

Com o escaneamento finalizado, é possível observar as vulnerabilidades encontradas, e com isso encontrar a forma mais assertiva para corrigi-las. O *Nessus* fornece os resultados em tela ou em forma de relatório com detalhes e classificação conforme ilustrado na Figura 11.

Fatecnológica
Jahu
ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

Figura 10 - Resultado do *Scan*



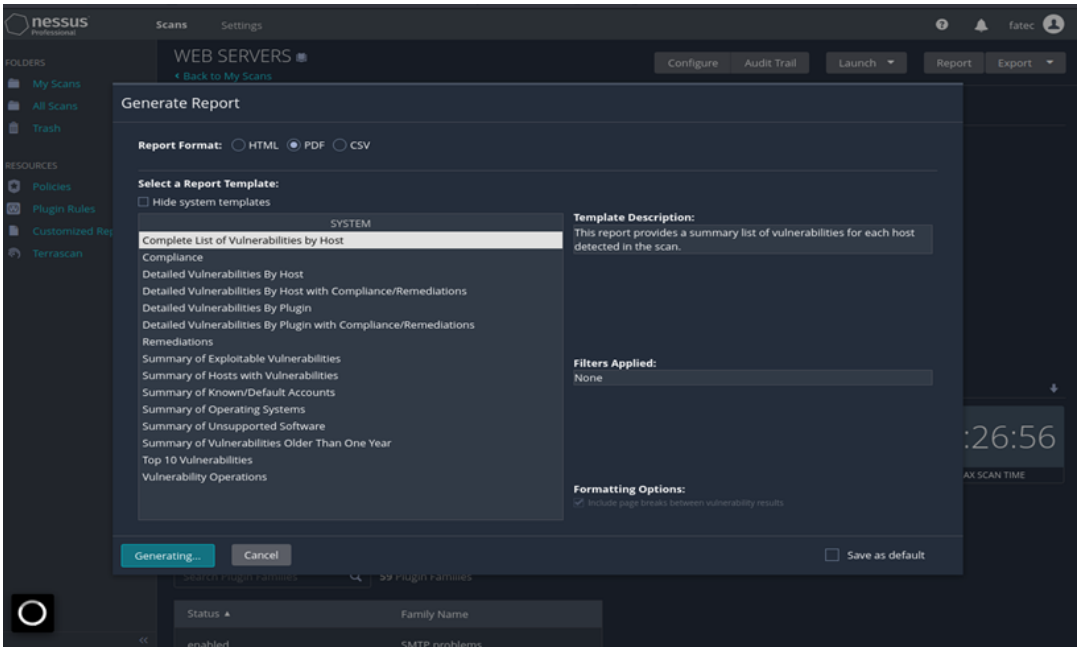
Fonte: Autoria Própria (2023)

Os filtros possibilitam a visualização e o foco em vulnerabilidades classificadas de acordo com diferentes níveis de severidade, como Crítica, Alta, Média, Baixa e Informacional. A severidade crítica refere-se a falhas de segurança que exigem correção imediata, uma vez que podem levar ao comprometimento total do sistema, vazamento de dados ou ao controle remoto por invasores.

O *Nessus* disponibiliza diversas opções de exportação de relatórios para auxiliar na análise e documentação de vulnerabilidades detectadas. A funcionalidade *Generate Report* permite criar relatórios que podem conter resumos detalhados de vulnerabilidades por *host*, verificações de conformidade e informações sobre falhas classificadas por *plugin*. Os filtros disponíveis possibilitam ajustar o conteúdo exibido, como destacar apenas vulnerabilidades exploráveis ou segmentar os sistemas operacionais afetados. Recursos como *Summary of Vulnerabilities Older Than One Year* são úteis para identificar riscos que persistem por longos períodos, enquanto as *Formatting Options* permitem a personalização do *layout* dos relatórios, assegurando uma apresentação clara e adaptada às necessidades da equipe de segurança, conforme mostrado na Figura 12.

Fatecnológica Jahu
ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

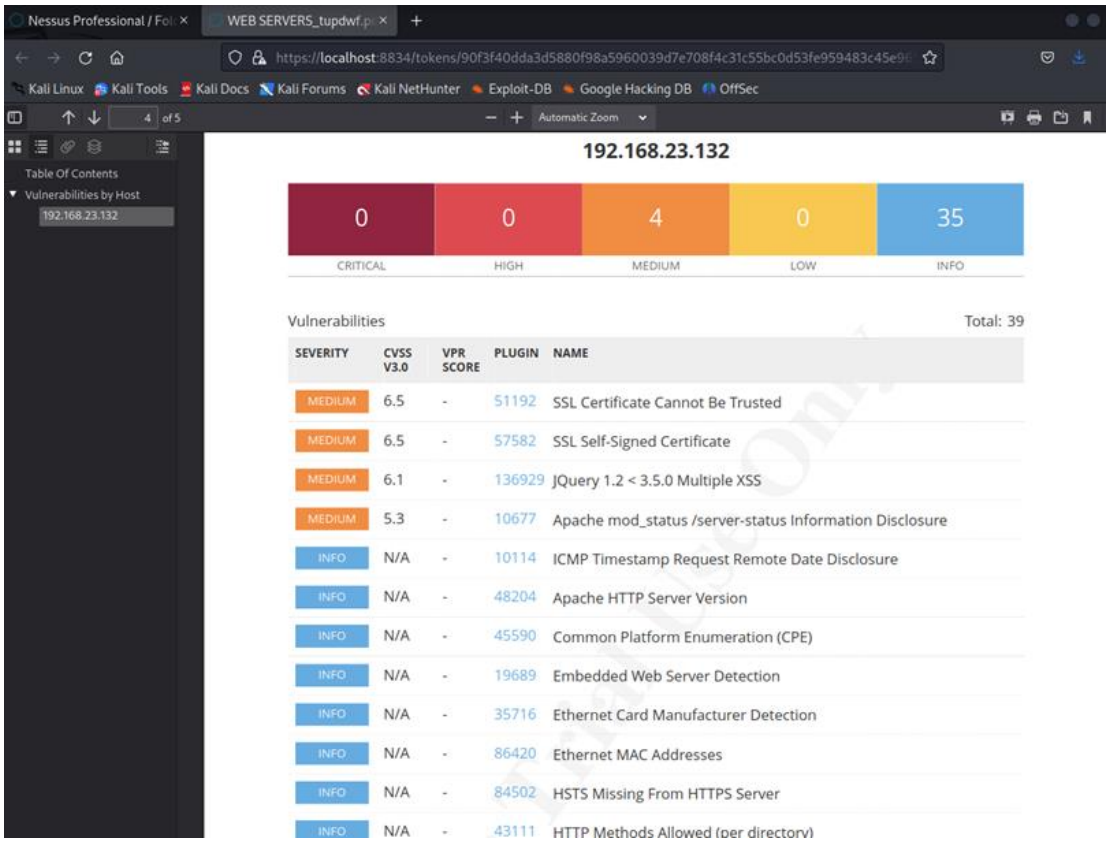
Figura 11 - Exportando relatórios



Fonte: Autoria Própria (2023)

Baseado nas informações do *Nessus* para o IP 192.168.23.132 revelou diversas vulnerabilidades, classificadas por severidade e impacto, conforme mostrado na Figura 13.

Figura 12 Resultados



Fonte: Autoria Própria (2023)

ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

Dentre as mais críticas, foi detectado que o certificado *SSL* do servidor não é confiável (*Plugin ID 51192*), o que pode abrir caminho para ataques de interceptação, como *Man-in-the-Middle* (MitM). Além disso, o uso de um certificado autoassinado (*Plugin ID 57582*) compromete a confiança em conexões seguras. A presença de uma versão desatualizada do *jQuery* (*Plugin ID 136929*) aumenta o risco de ataques de *Cross-Site Scripting* (XSS). Vulnerabilidades moderadas incluem a divulgação de informações pelo módulo *mod_status* do *Apache* (*Plugin ID 10677*) e a falta do cabeçalho *HTTP Strict Transport Security* (HSTS) (*Plugin ID 84502*), o que pode facilitar ataques de downgrade. Entre as descobertas informativas, estão a resposta a solicitações ICMP de timestamp (*Plugin ID 10114*) e a identificação da versão do Apache (*Plugin ID 48204*). Prioriza-se a correção das falhas críticas e de alta severidade, com uma análise adicional das vulnerabilidades informativas para decidir seu impacto e necessidade de mitigação.

Por tanto corrigir as vulnerabilidades críticas encontradas de alta severidade imediatamente, é altamente recomendável. Para vulnerabilidades informativas, recomenda-se uma análise para avaliar seu impacto no ambiente.

Nota-se também 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 (Barbosa; Ferreira; Neves, 2023).

Com a riqueza de informação que o *Nessus* traz, É possível 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 da segurança dos serviços online. Neste artigo, Foram explorados 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, destaca-se 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

ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

ameaças emergentes e com as ferramentas de apoio, como o *Nessus*, Obtém-se 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

BARBOSA, P.; FERREIRA, M.; NEVES, J. E. D. **Abordagem de Segurança no Desenvolvimento de Aplicações Web**. III FatecSeg. 2023. Disponível em: https://www.fatecourinhos.edu.br/fatecseg/index.php/fatecseg/article/view/107. Acesso em: 10 nov. 2024.

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

MOURA, T. M.; D' ALKMIN NEVES, J. E. **Análise de Segurança em Dispositivos Internet das Coisas**. Revista Interface Tecnológica, [S. l.], v. 18, n. 2, p. 15–27, 2021. Disponível em: https://doi.org/10.31510/infa.v18i2.1174. Acesso em: 10 nov. 2024.

NEVES, J. E. D. A. **Mineração de dados aplicada a simulação de cenários complexos em sistemas multiagentes**. Orientadores: Paulo Sérgio Martins Pedro (*in memoriam*), Marli de Freitas Gomes Hernandez. 2024. 237 p. Tese (Doutorado em Tecnologia) - Faculdade de Tecnologia, Universidade Estadual de Campinas (UNICAMP), Limeira, 2024. Disponível em: https://www.repositorio.unicamp.br/acervo/detalhe/1395946. Acesso em: 10 nov. 2024.

NEVES, J. E. D. A.; PEDRO, P. S. M.; HERNANDEZ, M. F. G.; FABRI JUNIOR, L. A. **Simulation of the Implementation of Domestic Solar Systems Using Multi-agent Systems from Web Scraping**. Smart Innovation, Systems and Technologies. 1ed.: Springer International Publishing, 2023, v. 1, p. 88-96. Disponível em: https://doi.org/10.1007/978-3-031-04435-9_8. Acesso em: 10 nov. 2024.

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

PRACIANO, E. **Como instalar e configurar o sudo no Debian**. Disponível em: https://elias.praciano.com/2015/11/como-instalar-e-configurar-o-sudo-no-debian. Acesso em: 15 jun. 2023.

SANTOS, A. M.; NEVES, J. E. D. **Exploração Maliciosa do ChatGPT para Ataques Cibernéticos**. III FatecSeg. 2023. Disponível em:

ISSN 1980-8526 (print)
ISSN 2763-5600 (online)

https://www.fatecourinhos.edu.br/fatecseg/index.php/fatecseg/article/view/108. Acesso em: 10 nov. 2024.

SOUZA, A. L. O.; BASTOS, C. V.; SANTOS, P. M. S.; SOARES, N. M.; NEVES, J. E. D. **Cibersegurança na Agricultura de Precisão:** Exploração à Aplicação de Medidas Preventivas. Advances in Global Innovation & Technology, v. 2, 61-73 p., 2024. Disponível em: https://doi.org/10.29327/2384439.2.2-5. Acesso em: 10 nov. 2024.

TENABLE. **Tenable - Nessus**. Disponível em: https://www.tenable.com/products/nessus. Acesso em: 14 jun. 2023.

TURNBULL, J. **Hardening Linux**. United States: Editora Apress, 2005.