Investigação de Vulnerabilidades em Aplicações Web utilizadas por Empresas de Leilões Online e Instituições de Ensino a Distância (EAD)

Hatanael Lima Fernandes[1], **Carlos Alberto da Silva**[2]

[1]Curso de Bacharelado em Sistemas de Informação
[2]Faculdade de Computação (FACOM)
[3]Universidade Federal de Mato Grosso do Sul (UFMS)
Av. Costa e Silva, s/nº – Bairro Universitário CEP: 79070-900 – Campo Grande – MS

hatanael.lima@ufms.br, carlos.silva@ufms.br

Abstract. *This article presents an analysis of security vulnerabilities found in web applications used by online auction companies and distance learning (EAD) institutions. The vulnerabilities were identified and assessed through pentesting techniques and automated tools, aiming to provide an overview of the security conditions of these platforms and to propose mitigation and prevention measures. The analysis revealed several flaws, highlighting the importance of proactive approaches to ensure system protection and prevent possible cyberattacks.*

Resumo. *Este artigo apresenta uma análise das vulnerabilidades de segurança presentes em aplicações web utilizadas por empresas de leilões online e instituições de ensino a distância (EAD). As vulnerabilidades foram identificadas e avaliadas por meio de técnicas de pentest e ferramentas automatizadas, com o objetivo de fornecer uma visão geral das condições de segurança dessas plataformas e propor medidas de mitigação e prevenção. A análise revelou a existência de diversas falhas, ressaltando a importância de abordagens proativas para garantir a proteção dos sistemas e evitar possíveis ataques cibernéticos.*

Palavras-chave: CVE, CWE, vulnerabilidades, aplicações web, pentest.

1. Introdução

A segurança cibernética destaca-se como um campo de importância estratégica crescente, diante da intensificação da dependência organizacional por sistemas digitais no desenvolvimento de suas atividades. No cenário atual, aplicações web tornaram-se ferramentas essenciais para empresas dos mais variados setores, incluindo as empresas de leilões online e instituições de Ensino a Distância (EAD), que lidam diariamente com dados sensíveis.

Considerando a relevância dessas aplicações, esta pesquisa tem como objetivo investigar vulnerabilidades presentes em sistemas web utilizados por tais organizações, a fim de compreender os riscos a que estão expostas e propor medidas que fortaleçam sua segurança. Por razões éticas e de confidencialidade, a identidade das instituições analisadas será mantida em sigilo.

A investigação foi conduzida com o apoio de ferramentas reconhecidas no campo da segurança da informação, como o Nmap, Gobuster, WhatWeb, WPScan, wafw00f e metasploit, focando na identificação de falhas exploráveis e na avaliação de sua criticidade. O estudo se baseou nos seguintes objetivos específicos: identificar vulnerabilidades nas aplicações web utilizadas, classificar os riscos conforme seu potencial de impacto e produzir um relatório técnico com recomendações para mitigação das ameaças.

Ao longo do estudo, foram analisados 29 sites pertencentes a empresas de leilões online e instituições de ensino a distância, utilizando técnicas de varredura e exploração controlada. Como resultado, identificaram-se 5 vulnerabilidades reais, evidenciando a exposição significativa dessas aplicações a riscos de segurança.

2. Fundamentação Teórica

Esta seção apresenta os principais conceitos e trabalhos relacionados que fundamentam teoricamente o desenvolvimento desta pesquisa. A análise de vulnerabilidades conduzida neste estudo baseia-se no sistema CVE Common Vulnerabilities and Exposures (CVE) [1], um repositório padronizado que reúne vulnerabilidades de segurança conhecidas publicamente. Mantido pela MITRE Corporation, com o apoio do Departamento de Segurança Interna dos Estados Unidos, o CVE tem como objetivo estabelecer uma nomenclatura comum que facilite a identificação, catalogação e o compartilhamento de informações sobre falhas de segurança.

De forma complementar, é adotado o padrão Common Vulnerability Scoring System (CVSS) [2], um sistema de pontuação que classifica a severidade das vulnerabilidades com base em diversas métricas técnicas, atribuindo a cada uma um valor numérico de 0 a 10. Essa padronização permite avaliar de maneira objetiva o impacto potencial das vulnerabilidades identificadas, favorecendo a priorização de medidas corretivas.

Além dos referenciais técnicos, esta pesquisa também se apoia em estudos anteriores que adotaram metodologias semelhantes. Destacam-se, por exemplo, os trabalhos ”Análise de vulnerabilidades em domínios WordPress: um estudo de caso em uma universidade pública”, de Gabriel Pastorello de Oliveira [3], e ”Verificando a segurança computacional do servidor de uma organização”, de Daniel Carvalho de Oliveira [4], que contribuíram significativamente para a definição da abordagem metodológica adotada nesta investigação.

3. Ferramentas

Nesta seção, são detalhadas as ferramentas que deram suporte à condução da investigação proposta.

3.1. Kali Linux

O Kali Linux [5] é uma distribuição Linux de código aberto baseada no Debian, anteriormente conhecida como BackTrack Linux. Essa distribuição permite que usuários realizem testes avançados de penetração e auditorias de segurança, sendo amplamente utilizada na área de segurança da informação. O Kali Linux inclui centenas de ferramentas, configurações e scripts voltados para atividades como computação forense, engenharia reversa e detecção de vulnerabilidades.

3.2. Nmap

O Nmap (Network Mapper) [6] é uma ferramenta de código aberto utilizada para exploração de redes e auditoria de segurança. Foi projetada para escanear rapidamente grandes redes, mas também funciona eficientemente em hosts individuais. O Nmap utiliza pacotes IP brutos para identificar quais hosts estão disponíveis na rede, quais serviços esses hosts oferecem, quais sistemas operacionais estão em execução, entre outras características.

3.3. Gobuster

Gobuster [7] é uma ferramenta utilizada para força bruta em diferentes alvos relacionados à segurança de redes e aplicações web. Entre suas funcionalidades estão a descoberta de URIs (diretórios e arquivos) em sites, subdomínios DNS (com suporte a curingas), nomes de Virtual Hosts em servidores web-alvo, além da identificação de *buckets* abertos na Amazon S3 e Google Cloud, e servidores TFTP abertos.

3.4. WhatWeb

WhatWeb [8] é uma ferramenta de código aberto para identificar tecnologias usadas em websites. Seu propósito é responder à pergunta “O que é aquele site?”. Ela detecta, por meio de *plugins*, componentes como CMS, plataformas de blog, pacotes de análise, bibliotecas *JavaScript*, servidores web e

dispositivos embarcados. Também é capaz de extrair informações como versões de software, e-mails, IDs de contas, módulos de frameworks web e mensagens de erro SQL. Com mais de 1.700 *plugins*, o WhatWeb é amplamente utilizado por profissionais de segurança para reconhecimento e mapeamento de tecnologias em auditorias e testes de penetração. Nesta pesquisa, foi utilizado com o objetivo de identificar tecnologias presentes nas aplicações analisadas, auxiliando na escolha de ferramentas e estratégias de investigação.

3.5. WPScan

WPScan [9] é uma ferramenta de código aberto voltada para a análise de segurança de websites que utilizam o WordPress como sistema de gerenciamento de conteúdo (CMS). Desenvolvida especificamente para identificar vulnerabilidades relacionadas a essa plataforma, o WPScan é amplamente utilizado por profissionais de segurança da informação em auditorias e testes de penetração.

3.6. Wafw00f

wafw00f [10] é uma ferramenta de código aberto utilizada para detecção e identificação de WAFs (Web Application Firewalls) em servidores web. Desenvolvida com foco na fase de reconhecimento em testes de segurança, a ferramenta aplica uma série de heurísticas para determinar a presença e o tipo de *firewall* de aplicação instalado.

3.7. Metasploit

Metasploit Framework [11] é uma plataforma de código aberto amplamente utilizada para testes de penetração, desenvolvimento de exploits e verificação de vulnerabilidades em sistemas e aplicações. Desenvolvido pela Rapid7, o Metasploit oferece um ambiente robusto que permite a simulação de ataques reais com o objetivo de identificar e corrigir falhas de segurança antes que sejam exploradas por agentes maliciosos.

4. Metodologia

De acordo com o **X-Force Threat Intelligence Index** [12], relatório anual elaborado pela equipe IBM X-Force, que reúne e analisa dados de segurança cibernética coletados por meio de centenas de bilhões de eventos monitorados em redes de clientes, bem como dados extraídos da *dark web* [13], *honeypots* [14] e incidentes reais de resposta a ataques, a exploração de vulnerabilidades é o segundo maior vetor de ataques cibernéticos, ficando atrás apenas do *phishing* [15].

O principal objetivo deste trabalho é identificar vulnerabilidades em aplicações web, seguindo os passos abaixo:

1. **Definição do escopo e preparação do ambiente:** Inicialmente, definiu-se o escopo da análise, que compreendia em aplicações webs utilizadas por empresas de leilões online e instituições de Ensino a Distância (EAD). Para isso, foi criado um ambiente de testes com a instalação do Kali Linux em uma máquina virtual, onde as ferramentas retromencionadas foram configuradas e testadas.
2. **Varredura inicial com Nmap:** Foi realizada uma varredura de rede utilizando o *Nmap*, com o objetivo de identificar hosts ativos, serviços em execução e portas abertas, fornecendo a base para os testes subsequentes.
3. **Enumeração de diretórios e subdomínios com Gobuster:** Em seguida, utilizou-se o *Gobuster* para forçar a descoberta de URIs ocultas, como diretórios administrativos e arquivos sensíveis, além de possíveis subdomínios presentes nos alvos.
4. **Identificação de firewalls com wafw00f:** A ferramenta *wafw00f* foi utilizada para detectar a presença de Web Application Firewalls (WAF), permitindo ajustar as técnicas de análise de acordo com as defesas encontradas.

5. **Coleta de tecnologias com WhatWeb:** Para identificar tecnologias empregadas nas aplicações web, como CMSs, *frameworks*, bibliotecas *JavaScript* e servidores, foi utilizado o *WhatWeb*, que permitiu compreender a estrutura do ambiente analisado.
6. **Análise de segurança com WPScan:** Em casos em que o sistema identificado utilizava WordPress, aplicou-se o *WPScan* para detectar vulnerabilidades conhecidas em *plugins*, temas e na própria instalação do CMS, com base no banco de dados WPVDB.
7. **Validação de exploração com Metasploit:** As falhas identificadas nas etapas anteriores, que possuíam *exploits* disponíveis no repositório do *Metasploit*, foram validadas por meio do *framework*, permitindo confirmar a possibilidade real de exploração e avaliar os potenciais impactos de cada vulnerabilidade.
8. **Pesquisa de CVEs relacionadas:** Com base nos resultados obtidos pelas ferramentas utilizadas nas etapas anteriores, foi realizada uma pesquisa sistemática por vulnerabilidades conhecidas Common Vulnerabilities and Exposures (CVE) e Common Weakness Enumeration (CWE) associadas às tecnologias, serviços e possíveis falhas identificadas durante a análise.
9. **Classificação das vulnerabilidades:** As vulnerabilidades encontradas foram classificadas com base no sistema de pontuação Common Vulnerability Scoring System (CVSS), conforme divulgado pela National Vulnerability Database (NVD), atribuindo um valor numérico de 0 a 10 para indicar a criticidade de cada falha e orientar as prioridades para mitigação.
10. **Proposição de medidas de mitigação:** Por fim, com base nas vulnerabilidades identificadas, foram sugeridas ações corretivas e preventivas visando a redução dos riscos e a implementação de novas políticas de segurança para o servidor.

5. Resultados

Nesta seção, são apresentados os resultados obtidos a partir das análises realizadas seguindo a metodologia retromencionada, sobre os ambientes avaliados. As vulnerabilidades identificadas foram analisadas com base nas informações disponíveis em bancos de dados públicos reconhecidos, como o CVE.org (mantido pela MITRE) e o *Exploit Database*, entre outros repositórios especializados em segurança da informação. Posteriormente foram classificados com base no sistema Common Vulnerability Scoring System (CVSS), utilizando o National Vulnerability Database (NVD). A Tabela 1 resume as principais vulnerabilidades confirmadas, incluindo suas respectivas classificações de severidade.

CVE	DESCRIÇÃO	CVSS
CWE-307	Improper Restriction of Excessive Authentication Attempts	Médio (6.7)
CWE-284	Improper Access Control	Médio (5.3)
CVE-2017-5487	WordPress REST API User Enumeration Vulnerability	Médio (5.3)
CVE-2017-7529	Integer Overflow / Partial Content Disclosure	Médio (5.3)
CVE-2011-1427	Kodak InSite 5.5.2 - '/Pages/login.aspx?Language' Cross-Site Scripting	Médio (4.3)

Tabela 1. Tabela de vulnerabilidades encontradas

A seguir, serão apresentadas breves descrições das vulnerabilidades identificadas, incluindo suas possíveis consequências e as respectivas recomendações de mitigação.

5.1. CWE-307 Improper Restriction of Excessive Authentication Attempts

Essa vulnerabilidade ocorre quando um sistema não impõe limites adequados ao número de tentativas de autenticação, permitindo que atacantes realizem ataques de força bruta para adivinhar credenciais válidas [16]. Isso pode resultar em acesso não autorizado ao sistema, comprometendo dados e funcionalidades sensíveis.

Recomendação: Implementar mecanismos de limitação de tentativas (rate limiting), como *delays* exponenciais, bloqueio temporário de IPs e uso de CAPTCHA após falhas consecutivas [17].

5.2. CWE-284 Improper Access Control

Trata-se de uma falha no controle de acesso, em que usuários podem acessar ou modificar recursos para os quais não possuem as permissões adequadas [18]. Esse problema pode levar à exposição ou alteração indevida de dados sensíveis, comprometendo a integridade, confidencialidade e disponibilidade do sistema.
Recomendação: Adotar políticas de controle de acesso baseadas em papéis (RBAC) e realizar auditorias periódicas para garantir que os privilégios estejam corretamente atribuídos [19].

5.3. CVE-2017-5487 WordPress REST API User Enumeration Vulnerability

Essa vulnerabilidade afeta versões do WordPress anteriores à 4.7.1, nas quais a API REST expõe informações de usuários, como nomes de *login*, sem autenticação [20]. Um atacante pode utilizar essa falha para enumerar usuários válidos e facilitar ataques de força bruta ou *spear phishing*.
Recomendação: Atualizar o WordPress para uma versão posterior à 4.7.1 e utilizar *plugins* ou filtros que restrinjam o acesso à API REST apenas a usuários autenticados [21].

5.4. CVE-2017-7529 Integer Overflow / Partial Content Disclosure

Presente em servidores Nginx com suporte a cabeçalhos de resposta parciais (*range requests*), essa vulnerabilidade permite que um atacante manipule valores de cabeçalho HTTP para causar um estouro de inteiros [22]. Como consequência, é possível vazar partes da memória do servidor, revelando informações sensíveis.
Recomendação: Atualizar o servidor Nginx para uma versão corrigida, além de considerar a desativação do suporte a range requests quando não for necessário [23].

5.5. CVE-2011-1427 Kodak InSite Multiple Cross-Site Scripting (XSS) Vulnerabilities

Essa vulnerabilidade afeta o Kodak InSite versão 5.5.2, na qual múltiplas falhas de *Cross-Site Scripting* (XSS) permitem que atacantes remotos injetem scripts ou HTML arbitrários via o parâmetro `Language` na página `Pages/login.aspx`, o parâmetro `HeaderWarning` na página `Troubleshooting/DiagnosticReport.asp` e o cabeçalho HTTP `User-Agent` na página `troubleshooting/speedtest.asp` [24]. A exploração pode levar à execução de código *JavaScript* malicioso no navegador das vítimas, possibilitando roubo de cookies, sequestro de sessão e outras ações maliciosas.
Recomendação: Aplicar atualizações ou mitigação recomendadas pelo fornecedor, validar e sanitizar entradas de usuários corretamente para prevenir a injeção de scripts maliciosos [25].

6. Conclusão

A realização deste trabalho possibilitou a análise prática da segurança em aplicações web por meio da identificação de vulnerabilidades reais em ambientes acessíveis publicamente. A utilização de ferramentas especializadas em testes de intrusão, como Nmap, Gobuster, whatweb, WPScan e outras soluções de varredura automatizada, permitiu mapear falhas e comportamentos inseguros com base em padrões reconhecidos, como as bases de dados CVE e classificações CVSS.

Os testes reforçam a importância de uma abordagem proativa na segurança de aplicações web, com auditorias regulares, correções contínuas e políticas de atualização de sistemas.

Conclui-se que o uso de ferramentas de *pentest* aliado a uma análise crítica dos resultados é uma etapa fundamental na proteção de aplicações web, contribuindo diretamente para a melhoria da postura de segurança da organização e para a mitigação eficaz de ameaças cibernéticas.

7. Trabalhos Futuros

Como desdobramento deste trabalho, futuras pesquisas podem explorar a análise mais aprofundada do impacto das vulnerabilidades identificadas, incluindo testes de exploração em ambientes controlados para avaliar cenários de ataque e suas consequências. Também é recomendável expandir a investigação para um número maior de aplicações web, abrangendo outros setores críticos que lidam com dados sensíveis, como saúde e finanças, a fim de verificar se os padrões de vulnerabilidades se repetem.

De modo complementar, pode-se aplicar metodologias de análise comparativa entre diferentes frameworks de desenvolvimento web, para identificar quais tecnologias apresentam maior propensão a falhas de segurança. Outro possível direcionamento é o desenvolvimento de um modelo automatizado de varredura e classificação de riscos, com base nos achados de CVEs, que auxilie organizações a identificar e corrigir falhas proativamente. Por fim, recomenda-se a análise de mecanismos de proteção existentes, como Web Application Firewalls (WAF), para verificar sua eficácia na mitigação das vulnerabilidades encontradas.

Referências

[1] MITRE Corporation. CVE - FAQs and resources. Disponível em: https://www.cve.org/About/Overview. Acesso em: 25 jun. 2025.

[2] National Institute of Standards and Technology. CVSS metrics – National Vulnerability Database. Disponível em: https://nvd.nist.gov/vuln-metrics/cvss. Acesso em: 25 jun. 2025.

[3] Gabriel Pastorello de Oliveira e Carlos Alberto da Silva. "Análise de vulnerabilidades em domínios WordPress: um estudo de caso em uma universidade pública". Trabalho de Conclusão de Curso. Universidade Federal do Mato Grosso do Sul - UFMS, 2023.

[4] Daniel Carvalho de Oliveira e Carlos Alberto da Silva. "Verificando a segurança computacional do servidor de uma organização". Trabalho de Conclusão de Curso. Universidade Federal do Mato Grosso do Sul - UFMS, 2025.

[5] Offensive Security. Kali Linux - Frequently Asked Questions (FAQ). Disponível em: https://www.kali.org/faq/. Acesso em: 25 jun. 2025.

[6] Gordon Lyon. Nmap reference guide. Disponível em: https://nmap.org/book/man.html. Acesso em: 25 jun. 2025.

[7] OJ Reeves. Gobuster - Directory/File DNS Busting Tool. Repositório GitHub. Disponível em: https://github.com/OJ/gobuster. Acesso em: 25 jun. 2025.

[8] WhatWeb. Ferramenta para identificação de tecnologias web. Disponível no Kali Linux. URL: https://tools.kali.org/information-gathering/whatweb. Acesso em: 25 jun. 2025.

[9] WPScan. Ferramenta para segurança em WordPress. Disponível em: https://wpscan.com/. Acesso em: 25 jun. 2025.

[10] WAFW00F. Detector de firewall de aplicações web. Repositório GitHub. Disponível em: https://github.com/EnableSecurity/wafw00f. Acesso em: 25 jun. 2025.

[11] Metasploit Framework. Plataforma de testes de penetração. Disponível em: https://www.metasploit.com/. Acesso em: 25 jun. 2025.

[12] IBM. Vulnerability scanning – IBM Security. Disponível em: https://www.ibm.com/think/topics/vulnerability-scanning. Acesso em: 25 jun. 2025.

[13] Adam Volle. Dark web — Definition, The Onion Router, History, Examples. Disponível em: https://www.britannica.com/technology/dark-web. Acesso em: 25 jun. 2025.

[14] Klaus Steding-Jessen, Marcelo H. P. C. Chaves e Cristine Hoepers. Honeypots e Honeynets: Definições e Aplicações. Disponível em: https://www.cert.br/docs/whitepapers/honeypots-honeynets/. Acesso em: 25 jun. 2025.

[15] Matthew Kosinski. O que é phishing? Disponível em: https://www.ibm.com/br-pt/think/topics/phishing. Acesso em: 25 jun. 2025.

[16] MITRE. CWE-307: Improper Restriction of Excessive Authentication Attempts. Disponível em: https://cwe.mitre.org/data/definitions/307.html. Acesso em: 25 jun. 2025.

[17] OWASP Foundation. Blocking Brute Force Attacks. Disponível em: https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks. Acesso em: 25 jun. 2025.

[18] MITRE. CWE-284: Improper Access Control. Disponível em: https://cwe.mitre.org/data/definitions/284.html. Acesso em: 25 jun. 2025.

[19] National Institute of Standards and Technology (NIST). Access Control Guidelines. Disponível em: https://csrc.nist.gov/publications. Acesso em: 25 jun. 2025.

[20] NVD - National Vulnerability Database. CVE-2017-5487 - WordPress REST API User Enumeration. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2017-5487. Acesso em: 25 jun. 2025.

[21] WordPress.org. API REST Restrictions Plugin. Disponível em: https://wordpress.org/plugins/disable-json-api/. Acesso em: 25 jun. 2025.

[22] NVD - National Vulnerability Database. CVE-2017-7529 - Nginx Integer Overflow Vulnerability. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2017-7529. Acesso em: 25 jun. 2025.

[23] NGINX. Security Advisories. Disponível em: https://nginx.org/en/security_advisories.html. Acesso em: 25 jun. 2025.

[24] MITRE Corporation. CVE-2011-1427 - Kodak InSite Multiple Cross-Site Scripting Vulnerabilities. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2011-1427. Acesso em: 25 jun. 2025.

[25] OWASP Foundation. Cross-Site Scripting (XSS) Prevention Cheat Sheet. Disponível em: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html. Acesso em: 25 jun. 2025.