XXI Escola Regional de Redes de Computadores / VIII Workshop de Segurança (ERRC/WRSeg). DOI: 10.5753/errc.2024.4689

ARTIGO COMPLETO/FULL PAPER

Análise empírica e comparativa de ferramentas de varredura de vulnerabilidades em aplicações Web usando OWASP BWA e Juice Shop

Empirical and Comparative Analysis of Vulnerability Scanning Tools in Web Applications Using OWASP BWA and Juice Shop

Ricardo Rosa • ricardorosa.aluno@UNIPAMPA.edu.br
Universidade Federal do Pampa (UNIPAMPA)
Diego Kreutz • diegokreutz@UNIPAMPA.edu.br
Universidade Federal do Pampa (UNIPAMPA)
Marcelino Garcia • marcelinogarcia.aluno@UNIPAMPA.edu.br
Universidade Federal do Pampa (UNIPAMPA)
Santiago Pereira • santiagopereira.aluno@UNIPAMPA.edu.br
Universidade Federal do Pampa (UNIPAMPA)
Rodrigo Mansilha • rodrigomansilha@UNIPAMPA.edu.br
Universidade Federal do Pampa (UNIPAMPA)

RESUMO. Neste artigo, realizamos uma análise empírica e comparativa das ferramentas de varredura de vulnerabilidades GoLismero, Nikto, Nuclei, OpenVAS, SecretScanner, Wapiti e ZAP, utilizando como ambientes de teste as aplicações reconhecidas OWASP Broken Web Applications (BWA) e Juice Shop. Nosso objetivo foi avaliar a eficácia e a abrangência da cobertura de vulnerabilidades detectadas por cada ferramenta. Os resultados indicam que a combinação de múltiplas ferramentas é essencial para alcançar uma cobertura mais ampla e eficiente, proporcionando maior proteção contra vulnerabilidades e ameaças cibernéticas.

ABSTRACT. In this article, we conduct an empirical and comparative analysis of the vulnerability scanning tools GoLismero, Nikto, Nuclei, OpenVAS, SecretScanner, Wapiti, and ZAP, using the recognized OWASP Broken Web Applications (BWA) and Juice Shop as test environments. Our objective was to evaluate the effectiveness and scope of vulnerability coverage provided by each tool. The results indicate that combining multiple tools is essential to achieve broader and more efficient coverage, offering greater protection against vulnerabilities and cyber threats.

PALAVRAS-CHAVE: Aplicação Web • Vulnerabilidades • Ferramentas de varredura • OWASP Top 10
KEYWORDS: Web Application • Vulnerabilities • Scanning Tools • OWASP Top 10

1 Introdução

Nos últimos anos, a digitalização e o aumento de aplicações em nuvem têm ampliado as vulnerabilidades de segurança nas organizações. A migração de programas tradicionais para versões web frequentemente negligencia a segurança, expondo dados pessoais e corporativos[1]. Essas falhas comprometem a integridade, confidencialidade e disponibilidade das informações, gerando riscos financeiros e de reputação[2]. Estima-se que 49% das aplicações web contenham vulnerabilidades graves, enquanto 13% estão expostas a falhas de segurança, destacando a necessidade urgente de mecanismos de defesa mais adequados[3, 4].

A detecção precoce de vulnerabilidades é essencial para evitar que invasores explorem falhas, reduzindo os custos com correções emergenciais. Ferramentas de caixa preta (*blackbox tools*) automatizam essa tarefa, identificando falhas como configurações incorretas, problemas de autenticação e injeções de código[5, 6]. Ao simular tentativas controladas de ataque, essas ferramentas ajudam a identificar possíveis vulnerabilidades antes que sejam exploradas[7]. Isso economiza tempo e recursos, permitindo que organizações tomem medidas preventivas de segurança.

Existem diversas ferramentas de varredura de vulnerabilidades disponíveis, tanto comerciais quanto de código aberto. Muitas operam em diferentes camadas de segurança e podem estar desatualizadas, sem suporte ou apresentar ineficiências, criando desafios para

Ricardo, Marcelino e Santiago são alunos e Diego e Rodrigo são professores do Programa de Pós Graduação em Engenharia de Software da UNIPAMPA.

especialistas[8, 9]. Dada a sofisticação crescente das ameaças, confiar em uma única ferramenta pode ser insuficiente para garantir uma proteção adequada. Estudos indicam que a combinação de diferentes ferramentas pode oferecer uma cobertura de segurança mais ampla e eficaz.[2, 10].

Diversos estudos avaliam a eficácia das ferramentas de varredura de vulnerabilidades. Um estudo recente indica OWASP ZAP e Nikto como as mais eficazes na detecção de falhas específicas [11]. Outros estudos destacam Nuclei e OpenVAS pela sua eficácia na identificação de vulnerabilidades como injeções e XSS [11, 5]. No entanto, esses trabalhos enfrentam limitações em termos da variedade e confiabilidade das aplicações alvo testadas, o que dificulta a comparação entre *scanners* em termos de cobertura e eficácia.

Neste estudo, realizamos uma análise comparativa de sete ferramentas de varredura de vulnerabilidades de código aberto: GoLismero, Nikto, Nuclei, OpenVAS, SecretScanner, Wapiti e ZAP. Diferente de estudos anteriores, além de utilizarmos um grupo maior de *scanners*, utilizamos as aplicações OWASP Broken Web Applications (BWA) e Juice Shop como ambientes de teste, ambas reconhecidas na área de segurança web [12, 13]. Avaliamos a eficácia dessas ferramentas na detecção de vulnerabilidades, buscando identificar combinações que proporcionem uma cobertura mais completa de vulnerabilidades.

1.1 Trabalhos Relacionados

A Tabela 1 resume os principais estudos relacionados que realizaram testes práticos, destacando as ferramentas utilizadas e as aplicações alvo. A análise da literatura mostra que muitos desses trabalhos avaliam um número limitado de *scanners* de vulnerabilidades. Além disso, a ausência de aplicações reconhecidas, como BWA e Juice Shop, reduz a abrangência dos estudos, dificultando a recomendação de combinações de ferramentas mais adequadas para diferentes cenários de segurança [3, 8]. A falta de justificativa clara para a escolha das ferramentas também é uma limitação recorrente.

Além dos estudos que realizaram testes práticos, outras pesquisas também contribuem significativamente para o campo da segurança em aplicações web. Vários surveys fornecem revisões sistemáticas da literatura sobre *scanners* de vulnerabilidades, abordando limitações, desafios e futuras direções [15, 10, 4, 5, 16]. Contudo, esses estudos carecem de uma conclusão definitiva sobre as melhores ferramentas de varredura e não utilizam de forma consistente aplicações-alvo como BWA e Juice Shop. Ademais, destacam a importância de testes contínuos, pois atualizações das ferramentas podem alterar seus resultados, tornando testes anteriores desatualizados.

Ref.	Ferramentas	Aplicações
[3]	Nessus, Nikto e Nmap	Cinco servidores web de Gana.
[2]	Acunetix, Burp Suite e OWASP ZAP	Duas aplicações web privadas.
[8]	Arachni, Burp Suite, BeEF, Ethercap, Hashcat, Nmap, SQLmap e OWASP ZAP	Uma aplicação customizada de E-library.
[12]	Burp Suite, bWAPP, WebGoat e Python	reddit.com
[14]	Burp Suite, Nmap, OWASP ZAP, Slowhttptest, Whois, WPScan e Zenmap	Um web site.

Tabela 1. Trabalhos relacionados: *scanners* de vulnerabilidades e aplicações alvo

Pesquisas destacam que aplicações web em áreas educacionais, governamentais e financeiras são frequentemente visadas por ataques devido à sua integração com sistemas de *backend*, como bancos de dados [16]. A seleção de ferramentas adequadas para avaliar vulnerabilidades é desafiadora, já que nenhuma ferramenta isolada cobre todas as falhas. Estudos anteriores (e.g., 2017) sugerem que a combinação de ferramentas, como W3AF e Nikto, pode aumentar a eficácia dos testes [2]. No entanto, essas conclusões podem estar desatualizadas, já que novas ferramentas com melhor cobertura e eficácia estão disponíveis atualmente.

2 Metodologia

Neste estudo, aplicamos técnicas de segurança ofensiva para identificar vulnerabilidades em aplicações web, focando na coleta de informações, reconhecimento do ambiente e detecção de falhas [17]. Utilizamos uma abordagem de caixa preta (*black-box*), simulando a perspectiva de um atacante externo [18]. As aplicações OWASP Broken Web Applications (BWA) e Juice Shop foram escolhidas como casos de estudo, e as ferramentas utilizadas seguiram critérios da lista *OWASP Vulnerability Scanning Tools* [19]. O uso do guia OWASP, especialmente o OWASP Top 10 de 2021 [20], foi fundamental para conduzir uma avaliação detalhada das vulnerabilidades mais críticas [14].

A metodologia adotou o *Vulnerability Assessment*

and Penetration Testing (VAPT), que identifica, quantifica e classifica falhas de segurança, utilizando ferramentas automatizadas para gerar relatórios e priorizar correções [4]. A combinação eficaz de ferramentas pode ajudar as empresas a reagir rapidamente e aumentar sua resiliência cibernética [21], sendo fundamental conhecer as táticas e ferramentas dos adversários para prevenir incidentes de segurança [18].

2.1 Seleção das Ferramentas de Varredura

Os *scanners* de vulnerabilidade para aplicações web, também conhecidos como ferramentas de *Dynamic Application Security Testing (DAST)*[1], são soluções automatizadas para identificar falhas de segurança, como injeções SQL, *cross-site scripting (XSS)*, injeções de comandos e configurações inseguras de servidores. Com uma ampla gama de opções, tanto comerciais quanto de código aberto, a escolha da ferramenta mais adequada exige uma avaliação criteriosa baseada nas necessidades de segurança específicas[19].

A OWASP oferece um catálogo abrangente com mais de 100 ferramentas de verificação de vulnerabilidades, embora não endosse avaliações externas, como as realizadas pelo projeto *Web Application Vulnerability Scanner Evaluation Project* (WAVSEP). Para este estudo, selecionamos ferramentas de código aberto compatíveis com plataformas Unix/Linux e testadas em nosso ambiente com Kali Linux. Inicialmente, consideramos ferramentas como Deepfence ThreatMapper, GoLismero, Grendel-Scan, Nikto, Nuclei, OpenVAS, OSTE Meta Scanner, purpleteam, Ride, SecretScanner, ThreatMapper, Vega, Wapiti e ZAP.

Cada ferramenta desempenha um papel relevante na análise de segurança, mas apresenta limitações. Por exemplo, o Deepfence ThreatMapper destaca-se no monitoramento e resposta a incidentes, mas enfrenta desafios como a complexidade de implementação, custos adicionais para recursos avançados, suporte técnico limitado e integração restrita com outras soluções. Além disso, sua curva de aprendizado elevada e relatórios pouco detalhados podem dificultar a adoção em organizações com menos recursos. Esses fatores são críticos para equilibrar custo-benefício e eficácia ao selecionar soluções de segurança para aplicações web.

Na segunda etapa da seleção de ferramentas, aplicamos três critérios principais: ausência de atualizações, eliminando ferramentas sem atualizações frequentes (última atualização anterior a 2020); presença de custos adicionais, removendo aquelas com versões gratuitas limitadas, onde a maioria dos recursos está disponível apenas em versões pagas; e indisponibilidade, desconsiderando ferramentas que não estavam mais disponíveis para download. Também foram excluídas ferramentas como Qualys Community Edition, devido à sua licença comunitária, e outras soluções comerciais, como Tenable Core + Nessus. Após esses filtros, as sete ferramentas selecionadas foram: GoLismero 2.0.3-1, Nikto 2.5.0, Nuclei 3.2.9, OpenVAS 23.4.1, SecretScanner 2.2.0, Wapiti 3.1.8 e ZAP 2.15.0.

2.2 Ambiente de Testes

Para configurar os testes, utilizamos o Oracle VirtualBox 7.0.14 com uma imagem do Kali Linux 2024.1, onde foram instaladas as sete ferramentas de varredura selecionadas. O ambiente foi configurado com 8 GB de RAM, 2 núcleos de processamento e rede NAT. A aplicação alvo Juice Shop 16.0.1 foi clonada do Docker Hub [2] e executada via Docker 2.27.0, em um ambiente controlado. Da mesma forma, foi baixada uma imagem no formato *Open Virtual Appliance (OVA)*[3] do BWA 1.2 por meio do SourceForge[4] e importada no VirtualBox. Essas configurações foram fundamentais para garantir testes eficientes e precisos na detecção de vulnerabilidades.

3 Resultados

Para cada aplicação testada, foi gerado um relatório detalhado com a identificação e validação das vulnerabilidades encontradas, incluindo sugestões de mitigação e ações corretivas, referenciadas ao OWASP Top 10. As vulnerabilidades foram classificadas em 13 grupos: arquivos potencialmente perigosos, arquivos sensíveis, ataques de injeção, autenticação e autorização, configuração de segurança, *cross-site scripting (XSS)*, detecção de serviços, *fingerprinting*, *headers* e configurações ausentes, informações divulgadas, informações privadas, protocolos SSH e segurança de protocolo.

A definição das 13 categorias de vulnerabilidades utilizadas neste estudo contou com o apoio da IA generativa, por meio do ChatGPT, que auxiliou na identificação e categorização de padrões. A categorização foi posteriormente validada e aprimorada com base no OWASP Top 10 de 2021.

Os dados foram visualizados em gráficos de bolhas nas Figuras 1 e 2, onde o eixo x representa o número de vulnerabilidades detectadas e o eixo y o número de categorias identificadas para cada ferramenta analisada.

[1] https://www.opentext.com/what-is/dast

[2] https://hub.docker.com/r/bkimminich/juice-shop

[3] https://docs.fileformat.com/pt/disc-and-media/ova/

[4] https://sourceforge.net/projects/owaspbwa/files/1.2/OWASP_

A análise comparativa revelou diferentes níveis de eficácia e cobertura nas aplicações OWASP Broken Web Applications (BWA) e Juice Shop. Os resultados, representados graficamente, destacam a variabilidade das capacidades de detecção das ferramentas em relação a 13 grupos de vulnerabilidades. No entanto, foi observada uma similaridade marcante entre as duas aplicações: ao comparar os gráficos lado a lado, constatou-se uma distribuição similar das ferramentas, com posições praticamente idênticas em ambas as aplicações.

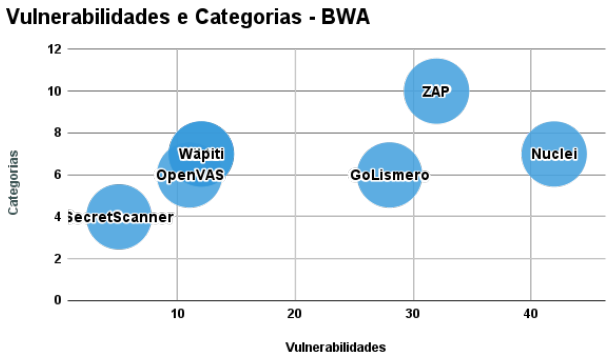

Figura 1. Vulnerabilidades e Categorias - BWA

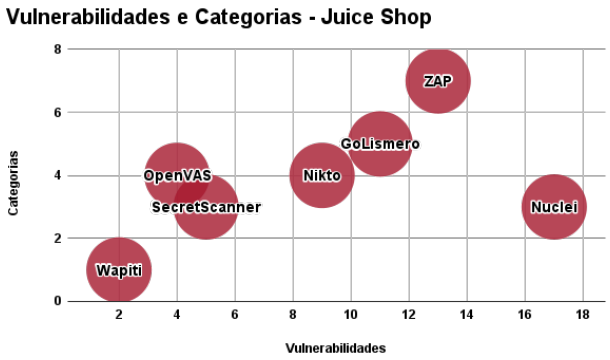

Figura 2. Vulnerabilidades e Categorias - Juice Shop

A Tabela 2 destaca que o Nuclei apresentou a maior cobertura total, detectando 59 vulnerabilidades. A ferramenta se destacou em segurança de protocolo, protocolo SSH e, especialmente, em *headers* e configurações ausentes, indicando eficácia na detecção de diferentes tipos de vulnerabilidades. Além disso, o Nuclei cobriu 61,54% das categorias de vulnerabilidades avaliadas, reforçando seu desempenho equilibrado em ambas as aplicações.

O ZAP também obteve resultados sólidos, detectando 45 vulnerabilidades e cobrindo 11 categorias (84,62%). Demonstrou alta eficácia na detecção de falhas em autenticação e autorização, além de um bom desempenho em técnicas de *fingerprinting*, tornando-o útil em cenários que requerem a identificação de características do sistema e verificação de controles de acesso.

O GoLismero destacou-se em categorias específicas, detectando 10 e 13 vulnerabilidades nas áreas de configuração de segurança e informações divulgadas, respectivamente. Isso demonstra que, enquanto algumas ferramentas oferecem cobertura geral mais ampla, outras podem proporcionar análises detalhadas em categorias específicas, dependendo do contexto dos testes de segurança. Em contraste, ferramentas como o SecretScanner detectaram um número menor de vulnerabilidades e apresentaram cobertura mais limitada nas aplicações BWA e Juice Shop, o que sugere que suas capacidades são mais restritas em comparação com ferramentas como Nuclei ou ZAP.

3.1 Discussão

Os resultados mostram que as ferramentas Nuclei e ZAP, quando combinadas, cobriram todas as 13 categorias de vulnerabilidades testadas no BWA, destacando sua relevância na análise de segurança web. No entanto, para o Juice Shop, foi necessário incluir outras ferramentas para uma cobertura mais completa, evidenciando a importância de uma abordagem combinada em diferentes ambientes de teste.

A análise sugere que a escolha e combinação de ferramentas devem ser orientadas pelos objetivos específicos dos testes de segurança, considerando as capacidades de cada uma. Ferramentas como Nuclei e ZAP oferecem cobertura abrangente, enquanto outras, como GoLismero e Nikto, são mais complementares em áreas específicas. Alinhar as necessidades de segurança com os pontos fortes das ferramentas contribui para aumentar a eficácia dos testes, proporcionando uma cobertura mais direcionada e precisa.

Observamos também que algumas vulnerabilidades não foram detectadas pelas ferramentas, indicando limitações em suas capacidades de cobertura. Uma análise mais aprofundada pode ajudar a identificar e explorar lacunas, especialmente ao considerar todas as vulnerabilidades reportadas. Isso é ilustrado pela página *Vulnerability Categories*[5] do Juice Shop, que, embora não tenha sido uma referência direta neste estudo, oferece insights para futuras pesquisas. Além disso, ao

Broken_Web_Apps_VM_1.2.ova/download

[5] https://pwning.owasp-juice.shop/companion-guide/latest/part1/categories.html

Tabela 2. Cobertura de Vulnerabilidades por Ferramenta nas Aplicações OWASP BWA e Juice Shop

Vulnerabilidades Categorizadas	Nuclei		ZAP		GoLismero		Nikto		OpenVAS		Wapiti		SecretScanner	
	BWA	JS	BWA	JS	BWA	JS	BWA	JS	BWA	JS	BWA	JS	BWA	JS
Arquivos Potencialmente Perigosos			2								1		1	
Arquivos Sensíveis	2	5	1	2	1		2	4						
Ataques de Injeção			3	1		2		1	2		1			
Autenticação e Autorização			7	1						1	1		1	1
Configuração de Segurança		1	3	3	10	3	4		1		4		1	
Cross-Site Scripting (XSS)			3		1	1			3		1			
Detecção de Serviços	5		4											
Fingerprinting	4		6	1			1			1	3	2		
Headers e Configurações Faltantes	11	11	2	3	2	1	2	2						
Informações Divulgadas	1			2	13	4	1	2	1	1				1
Informações Privadas			1		1		1		2				2	3
Protocolos SSH	9								2					
Segurança de Protocolo	10						1			1	1			
Vulnerabilidades	42	17	32	13	28	11	12	9	11	4	12	2	5	5
Vulnerabilidades (%)	29,58%	27,87%	22,54%	21,31%	19,72%	18,03%	8,45%	14,75%	7,75%	6,56%	8,45%	3,28%	3,52%	8,20%
Categorias (%)	53,85%	23,08%	76,92%	53,85%	46,15%	38,46%	53,85%	30,77%	46,15%	30,77%	53,85%	7,69%	30,77%	23,08%
Categorias (BWA + JS)	8 (61,54%)		11 (84,62%)		7 (53,85%)		8 (61,54%)		9 (69,23%)		7 (53,85%)		5 (38,46%)	

contrário do Juice Shop, o BWA não fornece um mapeamento claro de vulnerabilidades conhecidas, dificultando comparações precisas entre as aplicações e os *scanners* utilizados.

4 Conclusão

Neste estudo, destacamos a importância de combinar diferentes ferramentas de varredura de vulnerabilidades para obter uma cobertura mais abrangente em aplicações web. Identificamos que as ferramentas desempenham papéis complementares e que a integração de soluções diversas pode aumentar significativamente a eficácia dos testes de segurança. A combinação de uma ferramenta amplamente reconhecida, como o ZAP, com outras menos comuns, como Nuclei e GoLismero, mostrou-se uma abordagem promissora para alcançar uma cobertura mais completa e eficaz. O estudo também evidenciou que o uso isolado de uma única ferramenta é insuficiente para garantir uma segurança abrangente.

Concluímos que a escolha das ferramentas deve ser orientada pelos objetivos específicos dos testes e pelas características das aplicações analisadas. Uma abordagem integrada eleva a eficácia dos testes de segurança, proporcionando uma proteção mais consistente contra falhas e ameaças emergentes.

Declarações complementares

Financiamento

Esta pesquisa contou com o apoio da CAPES – Código de Financiamento 001.

Disponibilidade de dados e materiais adicionais

Os dados e/ou materiais adicionais poderão ser disponibilizados mediante solicitação.

Referências

1 Sampaio, F. F. Uma análise prática das principais vulnerabilidades em aplicações web baseado no top 10 OWASP, 2021.

2 Nagpure, S.; Kurkure, S. Vulnerability assessment and penetration testing of web application. *In*: IEEE. 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA). 2017. P. 1–6.

3 Appiah, V. *et al.* Survey of websites and web application security threats using vulnerability assessment. *Journal of Computer Science*, v. 15, n. 10, p. 1341–1354, 2018.

4 Ravindran, U.; Potukuchi, R. V. A Review on Web Application Vulnerability Assessment and Penetration Testing. *Review of Computer Engineering Studies*, v. 9, n. 1, 2022.

5 Curphey, M.; Arawo, R. Web application security assessment tools. *IEEE Security & Privacy*, v. 4, n. 4, p. 32–41, 2006. DOI: 10.1109/MSP.2006.108.

6 Khalid, M. N.; Rasheed, K.; Abid, M. M. *et al.* Web vulnerability finder (WVF): automated black-box web vulnerability scanner. *Int J Inf Technol Comput Sci*, v. 12, n. 4, p. 38–46, 2020.

7 Bertoglio, D. D. *et al.* Weasels e a construção de conhecimento em Segurança Ofensiva. *In*: SBC. ANAIS do II Simpósio Brasileiro de Educação em Computação. 2022. P. 109–117.

8 Holík, F.; Neradova, S. Vulnerabilities of modern web applications. *In*: IEEE. 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). 2017. P. 1256–1261.

9 Althunayyan, M. *et al.* Evaluation of black-box web application security scanners in detecting injection vulnerabilities. *Electronics*, MDPI, v. 11, n. 13, p. 2049, 2022.

10 Alazmi, S.; De Leon, D. C. A systematic literature review on the characteristics and effectiveness of web application vulnerability scanners. *IEEE Access*, IEEE, v. 10, p. 33200–33219, 2022.

11 Altulaihan, E. A.; Alismail, A.; Frikha, M. A survey on web application penetration testing. *Electronics*, MDPI, v. 12, n. 5, p. 1229, 2023.

12 Lakh, Y. *et al.* Investigation of the Broken Authentication Vulnerability in Web Applications. *In*: IEEE. 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). 2021. v. 2, p. 928–931.

13 Sinha, S. *Bug Bounty Hunting for Web Security*. Springer, 2019.

14 Pradhana, B. S. Website Security Analysis Using the OWASP10 Method (Case Study: almumtazparfumebatam. store). *Jurnal Kewarganegaraan*, v. 8, n. 1, p. 588–605, 2024.

15 Kritikos, K. *et al.* A survey on vulnerability assessment tools and databases for cloud-based web applications. *Array*, Elsevier, v. 3, p. 100011, 2019.

16 Nirmal, K.; Janet, B.; Kumar, R. Web application vulnerabilities-the hacker's treasure. *In*: IEEE. 2018 International Conference on Inventive Research in Computing Applications (ICIRCA). 2018. P. 58–62.

17 Amaral, É. *et al.* Unihacker: fundamentos da segurança I, 2021.

18 Alencar, I. D. d. *et al.* AuDiNoMiC: um gerenciador autonômico para auditorias em segurança ofensiva. Universidade Federal de Alagoas, 2019.

19 OWASP. *Vulnerability Scanning Tools*. 2024. https://owasp.org/www-community/Vulnerability_Scanning_Tools. https://owasp.org/www-community/Vulnerability_Scanning_Tools.

20 OWASP. *OWASP Top Ten*. 2024. https://owasp.org/www-project-top-ten/. https://owasp.org/www-project-top-ten/.

21 Lazarov, W. *et al.* Penterep: Comprehensive Penetration Testing with Interactive Checklists. *Available at SSRN 4743158*.