Sob a Lupa Corporativa Brasileira: Avaliação da Cobertura de Scanners de Vulnerabilidades em Aplicações Atuais

Thiago Paim Escarrone[1,2], **Ricardo Lazzari da Rosa**[2], **Diego Kreutz**[2], **Rodrigo Brandão Mansilha**[2], **Douglas Poerschke Rocha**[3,4]

[1]ANONIMIZADA (*a pedido da empresa*)

[2]Universidade Federal do Pampa (UNIPAMPA)

[3]DPR Consultoria em Tecnologia da Informação LTD

[4]Brasil TecPar

thiagoescarrone.aluno@unipampa.edu.br, ricardorosa.aluno@unipampa.edu.br,

diegokreutz@unipampa.edu.br, mansilha@unipampa.edu.br, douglas.poerschke@gmail.com

Resumo. *Neste estudo, apresentamos uma análise comparativa de scanners de vulnerabilidades para aplicações web, atendendo à demanda de três empresas parceiras que buscam avaliar a efetividade de soluções comerciais e gratuitas. Como principais contribuições, podemos destacar: a validação manual de todas as vulnerabilidades identificadas, uma avaliação abrangente de dez scanners (incluindo ferramentas já adotadas pelas empresas) com métricas de cobertura e precisão, e testes em ambientes diversificados como o OWASP Juice Shop e aplicações intensivas em JavaScript. Os resultados demonstram que tanto ferramentas gratuitas quanto comerciais apresentam limitações que devem ser consideradas pelas equipes técnicas, evidenciando a importância de estratégias que combinem múltiplas soluções e auditoria humana para garantir segurança efetiva em ambientes corporativos de produção.*

1. Introdução

A literatura sobre a eficácia de ferramentas de varredura de vulnerabilidades (*i.e.*, *scanners*) em aplicações web apresenta pelo menos três limitações importantes. Primeiro, grande parte das pesquisas concentra-se em contextos restritos, com ênfase em *scanners* aparentemente descontinuados ou que carecem de manutenção e suporte contínuos, como Sqlmap, Uniscan, XSSScan, RFuzz, XSSFuzz, RegFuzzer e Atropos [Khan et al. 2023, Shahid et al. 2022, Altulaihan et al. 2023, Güler et al. 2024]. A segunda limitação é a rara utilização de aplicações-alvo que podem ser consideradas *benchmarks* amplamente reconhecidos pela comunidade, como o OWASP Juice Shop. Essa omissão impede a generalização robusta dos resultados e dificulta a formulação de recomendações precisas sobre as melhores combinações de *scanners* para diferentes cenários de segurança cibernética [Appiah et al. 2018, Holík and Neradova 2017, Aydos et al. 2022, Zangana 2024, Kollepalli et al. 2024]. Por fim, a maioria dos trabalhos restringe sua análise a um número reduzido de *scanners* (tipicamente entre 4 e 6, majoritariamente gratuitos) [Shah 2020, Albahar et al. 2022, Idrissi et al. 2017]. O *survey* [Aydos et al. 2022] corrobora essas limitações, apontando a escassez de estudos focados em ferramentas e em atualização contínua de análises de segurança.

Em trabalho recente [Rosa et al. 2024], identificamos que *scanners* gratuitos isolados não oferecem cobertura adequada das vulnerabilidades em aplicações web estáticas, sendo necessário combinar múltiplas ferramentas para alcançar um grau satisfatório de detecção. Ampliando essa pesquisa, com o apoio de três empresas parceiras, realizamos neste estudo uma análise mais abrangente que inclui novas ferramentas gratuitas e soluções comerciais empregadas em ambientes corporativos reais. Além disso, passamos a considerar aplicações web dinâmicas no escopo da avaliação, o que evidenciou a limitada eficácia da maioria das ferramentas gratuitas diante de aplicações mais interativas e baseadas em *JavaScript*.

A avaliação comparativa entre soluções gratuitas e comerciais oferece subsídios relevantes para decisões estratégicas de cibersegurança, equilibrando cobertura, eficiência e custos operacionais[1]. Em síntese, este estudo avança em cinco frentes principais: (1) análise comparativa atualizada de dez *scanners* de vulnerabilidades; (2) validação manual de todas as vulnerabilidades para exclusão dos falsos positivos; (3) desenvolvimento e avaliação em uma aplicação *Single Page Application* criada especificamente para o estudo; (4) classificação da severidade das vulnerabilidades confirmadas através do *Common Vulnerability Scoring System (CVSS)*; e (5) unificação das vulnerabilidades para cálculo de *recall* relativo, permitindo comparação quantitativa do desempenho dos *scanners*. Especificamente, avaliamos oito ferramentas gratuitas (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, OWASP ZAP 23.4.1 e Qualys Community Edition 10.7.0-1) e duas soluções comerciais (Tenable Web App Scan 2.32.4-1790 e Burp Suite Professional v2025.4.4). Os testes foram conduzidos utilizando como alvos a aplicação estática OWASP Juice Shop e uma aplicação dinâmica desenvolvida para este estudo.

2. Metodologia e Ambiente de Testes

Este estudo adota uma abordagem de cibersegurança ofensiva, centrada na análise comparativa de ferramentas de varredura para aplicações web. Utilizou-se uma estratégia de teste de caixa-preta (*black-box*) conforme descrito por [Althunayyan et al. 2022] em sua avaliação de *scanners* de vulnerabilidades contra aplicações web modernas, simulando o comportamento de um atacante externo sem acesso prévio ao código-fonte. O objetivo foi avaliar a capacidade das ferramentas em detectar vulnerabilidades reais, com ênfase na cobertura (número e variedade de falhas identificadas), precisão (validação manual dos achados para eliminar falsos positivos[2]) e relevância prática (desempenho diante de características modernas, como rotas dinâmicas em SPAs)[Holík and Neradova 2017].

A condução do experimento seguiu uma sequência estruturada de etapas: primeiramente, selecionaram-se os *scanners* a serem avaliados; em seguida, configurou-se um ambiente de testes controlado para garantir uniformidade na execução dos testes e na exposição das aplicações vulneráveis. As varreduras foram executadas individualmente, com parâmetros padronizados para mitigar vieses nos resultados. Finalmente, cada vulnerabilidade reportada passou por validação manual, para confirmar sua veracidade e eliminar falsos positivos, elevando a confiabilidade dos resultados.

A seleção das ferramentas de varredura para este estudo foi pautada por critérios

[1] `https://www.wiz.io/academy/devsecops-tools/`

[2] `https://owasp.org/www-project-web-security-testing-guide/`

previamente definidos, focando na relevância prática e na aderência a ambientes corporativos. Consideramos aspectos como a frequência de atualizações, o tipo de licença (código aberto ou comercial), o grau de adoção na indústria e a compatibilidade com aplicações modernas. Dessa forma, incluímos soluções comerciais adotadas globalmente, como o Tenable WAS[3] e o Burp Suite Professional[4], que são referências no mercado e utilizadas por empresas de diversos setores ao redor do mundo. Assim, foi possível realizar uma análise comparativa abrangente de cobertura, precisão e aplicabilidade em cenários reais.

Para a avaliação das ferramentas, utilizamos três ambientes de testes distintos. As sete soluções de código aberto foram executadas em um ambiente virtualizado (Oracle VirtualBox 7.0.14 com Kali Linux 2024.1), configurado com 8GB de RAM e 2 núcleos de CPU. Paralelamente, o Burp Suite Professional operou em hardware dedicado (Windows 11, Intel Core i7 de 8ª geração, 32GB de RAM), simulando condições reais de análise corporativa para garantir resultados mais precisos. Por fim, as plataformas de *Software as a Service (SaaS)*[5], como Tenable WAS e Qualys Community Edition, foram acessadas diretamente via suas interfaces web.

Como alvo principal para os *scanners*, escolhemos a aplicação OWASP Juice Shop. Ela representa um cenário amplamente reconhecido em segurança web, sendo frequentemente utilizada como *benchmark* em estudos e treinamentos voltados à identificação e exploração de vulnerabilidades[6]. Uma instância atual do OWASP Juice Shop (versão 16.0.1) foi implantada no Google Cloud Run via container Docker[7], garantindo acessibilidade remota e ambiente controlado. Além disso, para desafiar os *scanners* com arquiteturas modernas, desenvolvemos uma aplicação Django/React que implementa padrões típicos de *Single Page Applications*, incluindo: (1) rotas de API dinâmicas consumidas via *JavaScript* e (2) renderização condicional de componentes. Essa abordagem se mostrou eficaz ao revelar limitações conhecidas [Holík and Neradova 2017] em ferramentas que não interpretam adequadamente contextos *JavaScript*, permitindo-nos avaliar a profundidade real da análise realizada por cada solução.

2.1. Classificação de Severidade com CVSS

A fim de padronizar a avaliação e a priorização das vulnerabilidades detectadas nesses alvos, adotamos o *Common Vulnerability Scoring System (CVSS)*. Reconhecido como um padrão aberto e fundamental para a avaliação quantitativa da severidade de falhas em sistemas computacionais, o CVSS não é apenas um guia, mas uma estrutura padronizada crucial para a indústria, que classifica falhas de segurança com base em métricas objetivas. Isso permite uma priorização inteligente e eficiente de riscos em ambientes corporativos, algo de suma importância para a tomada de decisão em segurança e sua gestão estratégica. Além disso, a literatura especializada tem consistentemente analisado o CVSS como uma métrica indispensável para aplicações em ambientes modernos e distribuídos [Almorsy et al. 2020], reafirmando seu valor prático e sua aplicabilidade no cenário industrial atual.

[3] `https://www.tenable.com/customers`
[4] `https://portswigger.net/customers`
[5] `https://shrtlink.ai/WhasIsSaaS`
[6] `https://shrtlink.ai/VulnerableApplicationsForPracticingPentesting`
[7] `https://console.cloud.google.com/`

Especificamente, a classificação de severidade CVSS 3.1 foi a metodologia central que norteou este estudo. Essa escolha permitiu (1) uma avaliação quantitativa padronizada, (2) a priorização inteligente de correções baseada no risco real e (3) comparações consistentes entre os *scanners*. Cada vulnerabilidade foi analisada segundo os vetores de métricas base do CVSS (explorabilidade e impacto técnico). A pontuação resultante, categorizada em cinco níveis de severidade (*Critical*, *High*, *Medium*, *Low* e *None*), reflete o potencial dano e a praticidade de exploração, criando um sistema de classificação multidimensional. Conforme demonstrado por [Janulevicius and Vasilecas 2017], essa abordagem supera as limitações de análises qualitativas, convertendo características técnicas em escores numéricos comparáveis, essencial para avaliar objetivamente o desempenho dos *scanners* em diferentes categorias de vulnerabilidade.

2.2. Eficácia dos scanners via Recall Relativo

Para complementar a classificação de severidade e medir a eficácia de cada *scanner* na identificação de vulnerabilidades, foi utilizada a métrica de *recall* relativo. Essa métrica compara o número de falhas detectadas por cada ferramenta com o total de vulnerabilidades confirmadas ao longo do estudo. Assim, ele fornece um parâmetro útil para avaliar e comparar o desempenho das soluções analisadas. A métrica é definida como:

$$\text{Recall Relativo} = \frac{\text{Vulnerabilidades Confirmadas pelo Scanner}}{\text{Total de Vulnerabilidades Confirmadas}} \tag{1}$$

Um *scanner* ideal teria um *recall* relativo de 1 (100%), o que indicaria a detecção completa de todas as vulnerabilidades confirmadas. Na prática, no entanto, mesmo as ferramentas mais avançadas raramente alcançam esse patamar. Isso ocorre devido a desafios inerentes a lógicas dinâmicas em *Single Page Applications* ou a vulnerabilidades contextuais, como falhas de autenticação. Os resultados obtidos nesta pesquisa ilustram bem essas variações e o desempenho de cada ferramenta.

3. Resultados

3.1. Desempenho geral

Na Figura 1, apresentamos o desempenho das ferramentas na detecção de vulnerabilidades, agrupadas nas oito categorias do estudo. Observamos que Tenable WAS e Qualys Community Edition alcançaram os melhores resultados, cobrindo a maioria das categorias e identificando um número significativo de vulnerabilidades. Ferramentas como o Wapiti tiveram um desempenho mais limitado, enquanto o OWASP ZAP e o Nuclei apresentaram resultados intermediários, o que aponta para variações na eficácia dos *scanners* testados. Ao comparar as soluções comerciais, o Tenable WAS demonstrou uma cobertura mais ampla e um volume maior de detecções, identificando vulnerabilidades em 7 das 8 categorias (87,5% de cobertura), contra 5 categorias (62,5%) do Burp Suite Professional. A diferença foi significativa no volume de vulnerabilidades confirmadas, com 61 achados válidos para o Tenable e 16 para o Burp Suite Professional.

Os resultados apontam que o Tenable WAS é ideal para grandes corporações e provedores de serviços de varredura devido à sua automação, ampla cobertura e integração

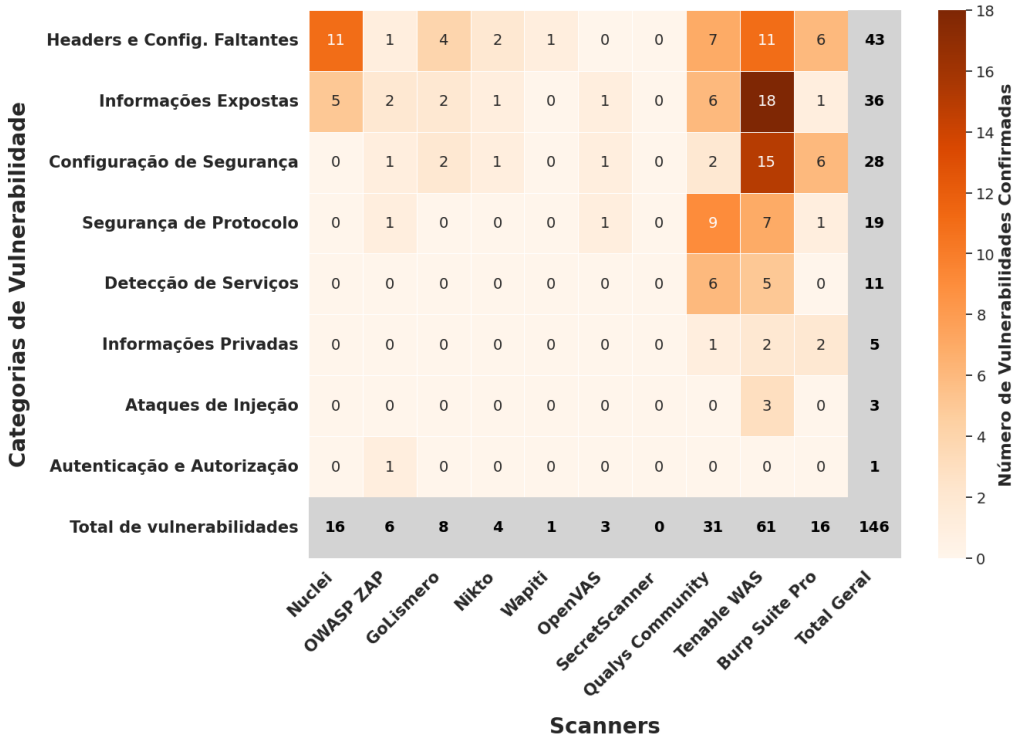

Figura 1. Número de vulnerabilidades confirmadas por categoria e scanner
Fonte: dados do estudo atual

com *pipelines* DevSecOps. Contudo, seu custo anual de R$ 30 mil[8] pode ser um obstáculo significativo para micro e pequenas empresas com orçamentos limitados.

Como alternativa, o Burp Suite Professional, com custo anual de aproximadamente R$ 2,5 mil[9], mostrou bom desempenho em análises pontuais. Para orçamentos restritos, uma abordagem híbrida combinando o Burp Suite Professional com ferramentas de código aberto como OWASP ZAP e Nuclei oferece uma solução equilibrada, preenchendo lacunas e proporcionando boa capacidade técnica e sustentabilidade financeira.

Por fim, o Qualys Community Edition destacou-se como uma ferramenta gratuita competitiva, confirmando 31 vulnerabilidades em 6 das 8 categorias analisadas, com um diferencial crucial na "Detecção de Serviços". Embora não tenha coberto "Autenticação e Autorização" (onde o OWASP ZAP se destacou), sua complementaridade com outras ferramentas gratuitas pode entregar resultados comparáveis a soluções comerciais, sendo uma opção estratégica para organizações com orçamentos limitados.

Ao analisar o impacto dos resultados, identificamos oportunidades para as empresas parceiras aprimorarem suas estratégias de segurança. Para a ANONIMIZADA, que usa o Tenable WAS, notamos lacunas em "Autenticação e Autorização" (coberta pelo OWASP ZAP) e falhas de SSL (detectadas pelo Qualys Community Edition); sugerimos complementar as varreduras com essas ferramentas gratuitas. Já para a DPR Consultoria, o Burp Suite Professional teve desempenho inferior ao Qualys Community Edition, e o Nuclei se mostrou mais eficaz na detecção de informações expostas que o Burp Suite.

[8]https://pt-br.tenable.com/products/web-app-scanning
[9]https://portswigger.net/burp/pro

3.2. Precisão e falsos positivos

A análise de falsos positivos (Figura 2) revelou que Tenable WAS e Burp Suite Pro tiveram apenas um falso positivo cada, demonstrando confiabilidade em suas abordagens distintas, enquanto o SecretScanner apresentou taxa de 100% ao reportar padrões genéricos sem verificação contextual. Em contraste, Nuclei e Wapiti não geraram falsos positivos, destacando-se pela precisão em suas detecções.

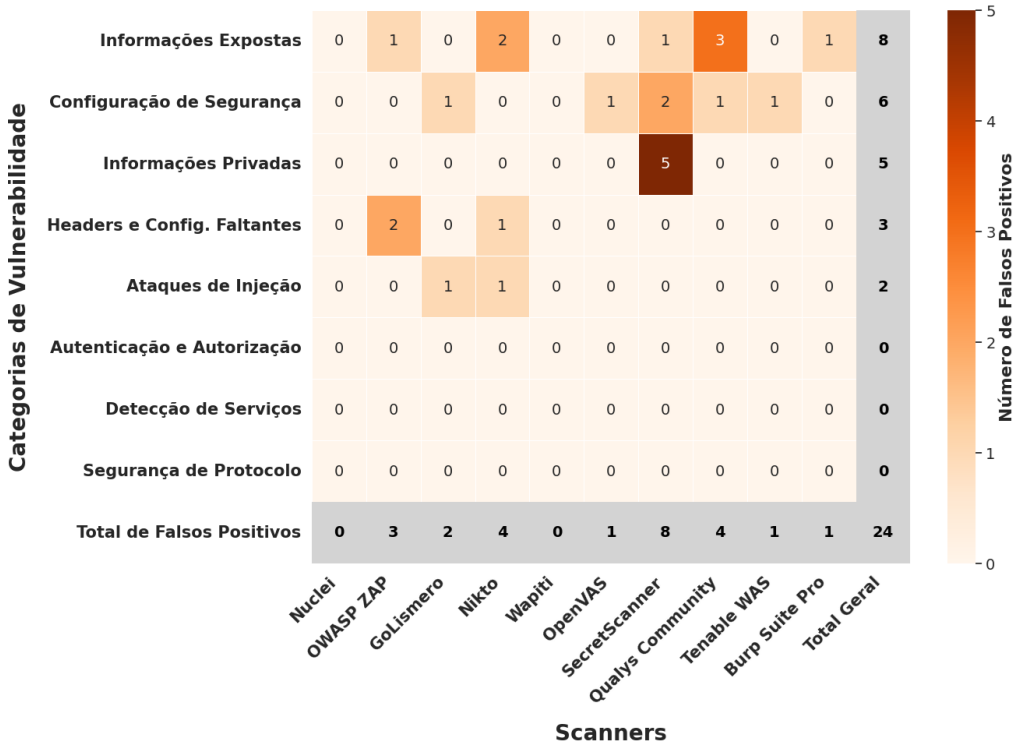

Figura 2. Número de falsos positivos por categoria e scanner
Fonte: dados do estudo atual

Em uma análise por categoria, "Autenticação e Autorização", "Detecção de Serviços" e "Segurança de Protocolo" não apresentaram falsos positivos. No entanto, a categoria "Informações Expostas" acumulou oito falsos positivos (33,33% do total de alertas incorretos). Esses dados sugerem que, embora algumas categorias possam ser verificadas de forma consistente com ferramentas automatizadas, outras exigem métodos adicionais de verificação, como análise manual e uso combinado de múltiplos *scanners*.

3.3. Eficácia dos Scanners

A análise dos da Tabela 1 resultados de *recall* relativo (detalhada na Seção 2.2) no Juice Shop revelou que o Tenable WAS (0.58) obteve o melhor desempenho entre as ferramentas pagas, enquanto o Burp Suite (0.15) apresentou resultado moderado, curiosamente igualado pelo Nuclei (0.15), demonstrando que soluções open-source podem atingir eficácia comparável a ferramentas comerciais em contextos específicos. O Qualys Community Edition (0.29) destacou-se entre as gratuitas, superando até mesmo o Burp Suite, enquanto Wapiti (0.01) e SecretScanner (0.00) não obtiveram resultados satisfatórios. Embora ferramentas pagas geralmente ofereçam melhor cobertura, algumas soluções gratuitas podem ser alternativas viáveis dependendo do cenário e configuração.

Tabela 1. Eficácia dos Scanners na Aplicação Juice Shop

Ferramenta	Juice Shop
Nuclei	0,15
OWASP ZAP	0,06
GoLismero	0,08
Wapiti	0,01
Qualys Community Edition	0,29
Nikto	0,04
Burp Suite Professional	0,15
Tenable WAS	0,58
SecretScanner	0,00
OpenVAS	0,03

3.4. Severidade das vulnerabilidades

A Figura 3 mostra que a maioria das vulnerabilidades detectadas são de severidade *Medium*, com distribuição inconsistente entre os *scanners*. Há disparidades notáveis na detecção de falhas *Critical/High*, principalmente em autenticação e injeção de código. Ferramentas como Tenable WAS e Qualys Community Edition focaram na abrangência (59% e 65% de achados *None*), o que aumenta os custos de filtragem de falsos positivos. Apenas quatro *scanners* (Tenable WAS, Burp Suite, OWASP ZAP e OpenVAS) identificaram vulnerabilidades *High*, e nenhuma detectou falhas *Critical* no Juice Shop. Isso revela sérias limitações das abordagens automatizadas para vulnerabilidades complexas que exigem análise contextual.

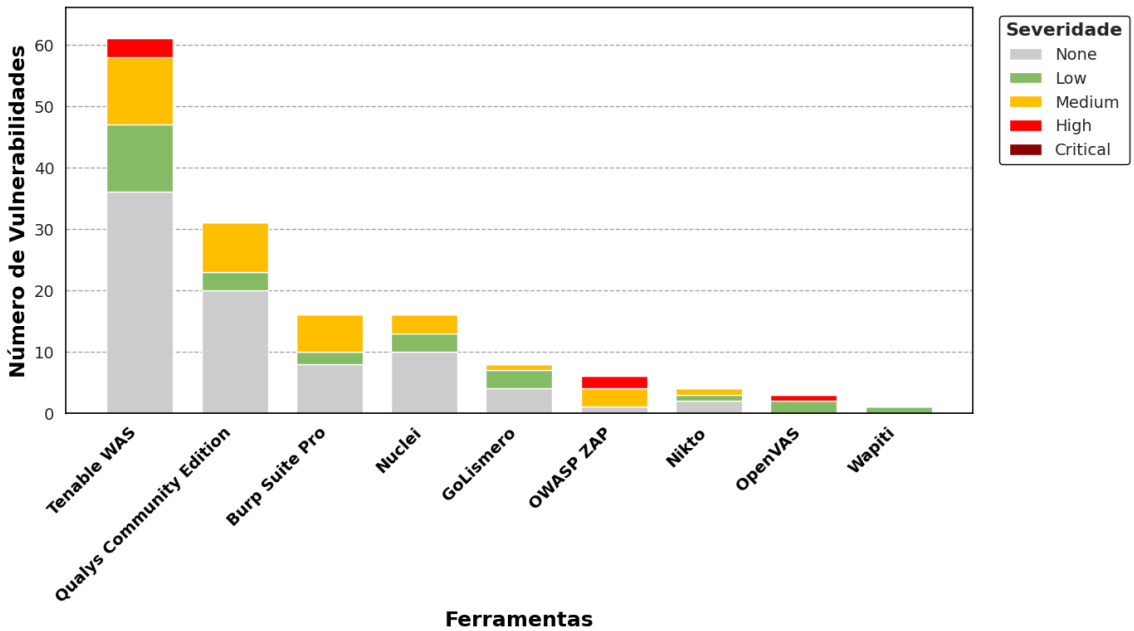

Figura 3. Severidade CVSS 3.1 em Juice Shop
Fonte: dados do estudo atual

3.5. SPAs e rotas dinâmicas

Nos testes com rotas dinâmicas *JavaScript*, apenas Tenable WAS, OWASP ZAP e Burp Suite Professional detectaram a API alvo. As demais ferramentas falharam por não interpretarem *JavaScript* ou simularem interações reais, limitando-se a HTML estático. Essa é uma deficiência que compromete a segurança de *Single Page Applications* e microserviços modernos, onde vulnerabilidades chave residem em *endpoints* assíncronos. Essa limitação demonstra a necessidade dos *scanners* evoluírem para abordagens dinâmicas que reproduzam fielmente o comportamento do usuário.

4. Conclusão e Trabalhos Futuros

Os resultados deste estudo demonstram que uma abordagem híbrida de *scanners* (Burp Suite Professional + OWASP ZAP + Nuclei) oferece uma cobertura complementar estatisticamente significativa. Ferramentas de código aberto, como OWASP ZAP e Nuclei, foram cruciais para identificar vulnerabilidades não detectadas por soluções comerciais, como falhas críticas de autenticação/autorização e maior eficácia em detecção de headers e configurações faltantes. Adicionalmente, o Qualys Community Edition superou o Tenable WAS na identificação de vulnerabilidades de segurança de protocolo. Em 37,5% das categorias analisadas, as detecções foram exclusivas de ferramentas gratuitas, validando a eficácia dessa estratégia híbrida, especialmente frente às limitações na detecção de APIs dinâmicas em arquiteturas SPA. Essa combinação metodológica é, portanto, essencial para uma avaliação de segurança robusta em ambientes corporativos/industriais heterogêneos.

Apontamos as seguintes direções para trabalhos futuros: (1) ampliar a avaliação para aplicações com arquiteturas diversificadas, incluindo cenários reais de maior complexidade; (2) incorporar ferramentas baseadas em IA para análise contextual de padrões complexos [Shar and Tan 2021]; e (3) adotar métricas padronizadas como o CVSS para uma priorização mais eficiente de vulnerabilidades em ambientes de produção.

Agradecimentos

Esta pesquisa contou com o apoio da CAPES (Código de Financiamento 001) e da FAPERGS por meio dos editais 02/2022, 08/2023 e 09/2023 (termos de outorga 22/2551-0000841-0, 24/2551-0001368-7 e 24/2551-0000726-1).

Referências

Albahar, M., Alansari, D., and Jurcut, A. (2022). An empirical comparison of pen-testing tools for detecting web app vulnerabilities. *Electronics*, 11(19).

Almorsy, M., Grundy, J., and Müller, I. (2020). An analysis of cvss-based vulnerability scores for cloud applications. *Journal of Systems and Software*, 170:110734.

Althunayyan, M., Saxena, N., Li, S., and Gope, P. (2022). Evaluation of black-box web application security scanners in detecting injection vulnerabilities. *Electronics*, 11(13):2049.

Altulaihan, E. A., Alismail, A., and Frikha, M. (2023). A survey on web application penetration testing. *Electronics*, 12(5).

Appiah, V., Asante, M., Nti, I. K., and Nyarko-Boateng, O. (2018). Survey of websites and web application security threats using vulnerability assessment. *Journal of Computer Science*, 15(10):1341–1354.

Aydos, M., Çiğdem Aldan, Coşkun, E., and Soydan, A. (2022). Security testing of web applications: A systematic mapping of the literature. *Journal of King Saud University - Computer and Information Sciences*, 34(9):6775–6792.

Güler, E., Schumilo, S., Schloegel, M., Bars, N., Görz, P., Xu, X., Kaygusuz, C., and Holz, T. (2024). Atropos: Effective fuzzing of web applications for {Server-Side} vulnerabilities. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 4765–4782.

Holík, F. and Neradova, S. (2017). Vulnerabilities of modern web applications. In *40th MIPRO*, pages 1256–1261. IEEE.

Idrissi, S. E., Berbiche, N., Guerouate, F., and Shibi, M. (2017). Performance evaluation of web application security scanners for prevention and protection against vulnerabilities. *International Journal of Applied Engineering Research*, 12(21):11068–11076.

Janulevicius, A. and Vasilecas, O. (2017). A comparison of vulnerability scoring systems for industrial web applications. *Computer Standards & Interfaces*, 54:50–57.

Khan, B., Bangash, J. I., Tariq, M., Gul, N., Zahir, S., and Kamal, A. (2023). A comparative model to analyze various web application penetration testing tools for different vulnerabilities. In *ICTAPP*, pages 1–6.

Kollepalli, R. P. K., Reddy, M. J. S., Sai, B. L., Natarajan, A., Mathi, S., and Ramalingam, V. (2024). An experimental study on detecting and mitigating vulnerabilities in web applications. *International Journal of Safety & Security Engineering*, 14(2).

Rosa, R., Kreutz, D., Garcia, M., Pereira, S., and Mansilha, R. (2024). Análise empírica e comparativa de ferramentas de varredura de vulnerabilidades em aplicações web usando owasp bwa e juice shop. In *Anais da XXI Escola Regional de Redes de Computadores*, pages 183–188, Porto Alegre, RS, Brasil. SBC.

Shah, M. P. (2020). *Comparative analysis of the automated penetration testing tools*. PhD thesis, Dublin, National College of Ireland.

Shahid, J., Hameed, M. K., Javed, I. T., Qureshi, K. N., Ali, M., and Crespi, N. (2022). A comparative study of web application security parameters: Current trends and future directions. *Applied Sciences*, 12(8).

Shar, L. K. and Tan, H. B. K. (2021). Machine learning for security vulnerability detection: A survey. *Journal of Computer Security*, 29(3):301–351.

Zangana, H. M. (2024). Exploring the landscape of website vulnerability scanners: A comprehensive review and comparative analysis. *Redefining Security With Cyber AI*, pages 111–129.