XXI Escola Regional de Redes de Computadores (ERRC). DOI: 10.5753/errc.2024.4674

Artigo Completo/Full Paper

Classificação Multicritério de Ferramentas de Análise de Vulnerabilidades em Docker

Multi-Criteria Ranking of Docker Vulnerability Analysis Tools

Ali Iddar • ali.iddar@acad.ufsm.br
Universidade Federal de Santa Maria (UFSM)
Rogério C. Turchetti • turchetti@redes.ufsm.br
Universidade Federal de Santa Maria (UFSM)

Resumo. Este estudo avalia o desempenho de ferramentas de análise de vulnerabilidades em imagens Docker utilizando o Processo de Análise Hierárquica (AHP) como metodologia de tomada de decisão multicritério. Diante do crescente uso de contêineres Docker e dos riscos associados a vulnerabilidades presentes nas imagens disponíveis no Docker Hub, a pesquisa visa comparar e classificar as principais ferramentas destinadas à identificação dessas fragilidades. O AHP foi aplicado em duas etapas: inicialmente, em cada estudo individualmente e, posteriormente, aos resultados consolidados de todos os estudos. Os resultados indicaram que a escolha da ferramenta deve levar em conta, além da eficácia geral, fatores contextuais e a possibilidade de utilizar múltiplas ferramentas em paralelo para uma detecção mais precisa. A pesquisa destaca ainda a necessidade de avaliações contínuas, especialmente em relação às abordagens de análise dinâmica e suas comparações com ferramentas de análise estática.

Abstract. In this study, we evaluated the performance of vulnerability analysis tools for Docker images using the Analytic Hierarchy Process (AHP) as a multi-criteria decision-making methodology. Given the increasing use of Docker containers and the risks associated with vulnerabilities in the images available on Docker Hub, the research aims to compare and rank the main tools designed to identify these weaknesses. We applied the AHP method in two stages: initially, in each study, and later, we used it to the consolidated results of all studies. Our results indicated that the choice of tool should take into account not only overall effectiveness but also contextual factors and the possibility of using multiple tools in parallel for more precise detection. The research also highlights the need for continuous evaluations, especially regarding dynamic analysis approaches and their comparisons with static analysis tools.

Palavras-chave: Análise de vulnerabilidades • Docker • Segurança de Contêiners • Método AHP
Keywords: AHP Method • Container security • Docker • Vulnerability analysis

1 Introdução

O Docker emergiu como uma tecnologia revolucionária no campo da virtualização, oferecendo uma abordagem eficiente e flexível para o desenvolvimento, implantação e gerenciamento de aplicações. A plataforma Docker utiliza contêineres que compartilham o kernel do sistema operacional hospedeiro, resultando em uma solução mais leve e eficiente do que máquinas virtuais [1]. A popularidade do Docker cresceu rapidamente com o Docker Hub, repositório de imagens mantidas e gerenciadas por fornecedores certificados e pela comunidade [2]. Essa facilidade de compartilhamento, embora benéfica para a comunidade de desenvolvedores, também introduz riscos significativos à segurança.

O estudo de Shu, Gu e Enck (2017)[2] analisou 356.218 imagens no Docker Hub, revelando que tanto as imagens oficiais quanto as da comunidade contêm, em média, mais de 180 vulnerabilidades quando consideradas todas as versões. Além disso, mais de 80% das imagens, independentemente de sua origem, apresentam pelo menos uma vulnerabilidade de alta gravidade [2]. Esses dados destacam a necessidade crítica de uma abordagem mais rigorosa em relação à segurança das imagens usadas nos contêineres.

A segurança no contexto do Docker envolve múltiplos aspectos. É essencial garantir que as imagens estejam livres de vulnerabilidades e/ou configurações incorretas antes de sua distribuição. Isso inclui a varredura contínua em busca de *malware*, utilizando conjuntos de assinaturas e métodos de detecção usando heurística comportamental [3]. Além disso, práticas como o armazenamento seguro de segredos fora das imagens e a manutenção de um conjunto confiável de imagens e registros, são cruciais para mitigar riscos de segurança.

O ecossistema Docker apresenta desafios únicos de segurança devido à sua natureza. Os contêineres se comunicam diretamente com o kernel da máquina hospedeira, o que, embora eficiente, potencialmente

aumenta a superfície de ataque [4]. O Docker utiliza *namespaces* do Linux para isolamento, mas certos sistemas de arquivos do kernel permanecem não isolados, representando riscos adicionais. A complexidade do ambiente Docker é evidenciada pela diversidade de vulnerabilidades encontradas. O estudo de Martin, A. et al. [5] revelou que 36% das imagens oficiais no Docker Hub contêm vulnerabilidades *Common Vulnerabilities and Exposures* (CVE) de alta prioridade. Além disso, o movimento DevOps (Desenvolvimento e Operações), facilitado pelo Docker, pode inadvertidamente levar à inclusão de ferramentas de desenvolvimento ou versões de pacotes desatualizadas nas imagens, aumentando ainda mais os riscos de segurança.

Neste contexto, as ferramentas de análise de vulnerabilidades tornam-se fundamentais. Elas desempenham um papel vital na identificação e mitigação de riscos de segurança em imagens Docker. Essas ferramentas são capazes de realizar varreduras detalhadas, detectando vulnerabilidades conhecidas, configurações incorretas e até mesmo código malicioso potencial. A importância dessas ferramentas é amplificada pelo fato de que muitos usuários não estão cientes dos riscos associados às imagens que utilizam. O estudo Liu, P. et al. [6] mostrou que 97% dos usuários se preocupam apenas com a execução bem-sucedida da imagem, ignorando parâmetros sensíveis nos comandos de execução. Essa falta de conscientização ressalta a necessidade de ferramentas automatizadas e eficazes para análise de segurança. Além disso, o tempo médio para corrigir uma vulnerabilidade em imagens Docker é significativamente maior do que em software comum — 422 dias em comparação com 181 dias [6]. Esse atraso na correção de vulnerabilidades aumenta a janela de oportunidade para potenciais ataques, tornando ainda mais crítico o uso de ferramentas de análise de vulnerabilidades como parte integrante do ciclo de vida de desenvolvimento e implantação de contêineres.

Existem duas abordagens principais para a análise de vulnerabilidades em imagens Docker [7]: a análise estática e a análise dinâmica. A análise estática examina o conteúdo da imagem sem a executar, focando principalmente na inspeção do código-fonte, das dependências e das configurações. Ferramentas como Anchore, Trivy e Clair são exemplos de soluções que realizam análise estática. Essas ferramentas geralmente fazem a varredura das imagens em busca de vulnerabilidades conhecidas, comparando os pacotes e suas versões com bancos de dados de CVE [8].

Por outro lado, a análise dinâmica observa o comportamento do contêiner durante sua execução [7]. Essa abordagem permite a detecção de vulnerabilidades que só se manifestam em tempo de execução, como comportamentos maliciosos ou anomalias no uso de recursos. Técnicas de análise dinâmica em um dos estudos analisados utilizaram algoritmos de aprendizado de máquina não supervisionado para identificar padrões anômalos nas chamadas de sistema, no uso de recursos ou no tráfego de rede [8].

A principal diferença entre as duas abordagens reside na natureza e no momento da análise. Enquanto a análise estática oferece uma visão abrangente das vulnerabilidades potenciais antes da execução do contêiner, a análise dinâmica proporciona *insights* sobre o comportamento real do contêiner em tempo de execução [9]. A análise estática é geralmente mais rápida e menos intensiva em recursos, mas pode não detectar vulnerabilidades que só se manifestam durante a execução. Já a análise dinâmica, embora mais intensiva em recursos e potencialmente mais lenta [7], pode identificar ameaças que escapam à análise estática [8].

Ao longo dos últimos anos, foram desenvolvidas diversas ferramentas de análise de vulnerabilidades em imagens Docker, e diversos estudos foram realizados avaliando a eficácia dessas ferramentas. Este estudo visa realizar uma avaliação comparativa das ferramentas de análise de vulnerabilidades em imagens Docker, avaliando os resultados desses estudos e comparando-os através do uso da metodologia de Processo de Análise Hierárquica. Os estudos foram selecionados por meio de uma busca sistemática nas seguintes bases acadêmicas: IEEE Xplore, EI Compendex, Web of Science, ACM Digital Library, Scopus e Google Acadêmico. Foram encontradas 62 publicações, submetidas a um processo de seleção com base em critérios pré-definidos e estabelecidos na metodologia. Essa seleção resultou em 10 estudos que avaliaram as ferramentas em termos de eficácia de detecção de vulnerabilidades, como é mostrado na Tabela 1.

2 Metodologia

Para avaliar objetivamente a eficácia das ferramentas de detecção de vulnerabilidades em imagens Docker e eliminar as dificuldades impostas pelos múltiplos critérios de avaliação usados nos estudos analisados, adotou-se a abordagem de *Multi-Criteria Decision Making (MCDM)*[1]

[1] ***Tomada de Decisão Multicritério (MCDM)***: é um dos ramos mais conhecidos da teoria de tomada de decisão. Ele lida com problemas de decisão que envolvem múltiplos critérios, que podem estar em conflito entre si e ter unidades de medidas diferentes. Os critérios recebem pesos de importância e o problema pode ser representado em um formato chamado de matriz de decisão [19].

ID	Título	Ano
Pub-1	A Study on Container Vulnerability Exploit Detection [8]	2019
Pub-2	Container Vulnerability Scanners: An Analysis [10]	2020
Pub-3	An Analysis of Security Vulnerabilities in Container Images for Scientific Data Analysis [11]	2021
Pub-4	An Evaluation of Container Security Vulnerability Detection Tools [12]	2021
Pub-5	Segurança em Imagens Docker: Um Estudo de Ferramentas de Análise Estática [13]	2021
Pub-6	Concerns About Available Container Image Scanning Tools and Image Security [14]	2022
Pub-7	Investigating the Inner Workings of Container Image Vulnerability Scanners [15]	2022
Pub-8	Continuous Docker Image Analysis and Intrusion Detection Based on Open-Source Tools [16]	2022
Pub-9	Vulnerability Analysis of Docker Hub Official Images and Verified Images [17]	2023
Pub-10	Detecting Container Vulnerabilities Leveraging the CI/CD Pipeline [18]	2023

Tabela 1. Estudos selecionados na revisão sistemática.

especificamente o método *Analytic Hierarchy Process (AHP)*.

O Processo de Análise Hierárquica (AHP), desenvolvido por Thomas L. Saaty em 1970 [20], destaca-se por sua capacidade de abordar problemas complexos com múltiplos critérios de avaliação. A principal vantagem é estruturar problemas hierárquicamente, permitindo a avaliação sistemática de fatores quantitativos e qualitativos [21]. No contexto deste trabalho, o AHP é particularmente útil devido à complexidade do problema, e sua capacidade de incorporar julgamentos pessoais de forma estruturada [22].

Para este estudo, o método AHP foi adotado para classificar as ferramentas de análise de vulnerabilidades em uma escala de 1 a 5 estrelas, considerando não apenas as métricas de desempenho, mas também a quantidade de estudos realizados sobre cada ferramenta. Essa abordagem visa proporcionar uma classificação mais robusta e confiável. O método foi aplicado em duas etapas: (1) aplicação individual em cada estudo e (2) aplicação nos resultados consolidados. Cada etapa seguiu os passos: (a) montagem da matriz de julgamento, (b) cálculo dos pesos, (c)verificação da consistência e (d) classificação das ferramentas.

Para facilitar e agilizar a aplicação do método AHP, foram desenvolvidas planilhas que estão disponíveis no repositório GitHub [2]. Essas planilhas realizam os cálculos complexos do AHP automaticamente, reduzindo o tempo e esforço na análise. As planilhas abrangem as principais etapas do método, oferecendo vantagens como redução de erros de cálculo, economia de tempo, padronização do processo AHP.

3 Resultados e Discução

A avaliação e classificação das ferramentas de análise de vulnerabilidades foi realizada através da aplicação do Processo de Análise Hierárquica (AHP) em duas etapas distintas. Na primeira etapa, o método AHP foi aplicado individualmente para cada publicação analisada, considerando as métricas específicas utilizadas em cada estudo. A Figura 2 apresenta um resumo visual dos resultados dessa aplicação inicial do AHP, mostrando a pontuação obtida por cada ferramenta em cada estudo analisado. Na segunda etapa, os resultados consolidados de todos os estudos foram submetidos a uma nova aplicação do AHP. Esse processo resultou em uma classificação geral das ferramentas em uma escala de 1 a 5 estrelas, conforme ilustrado na Figura 1.

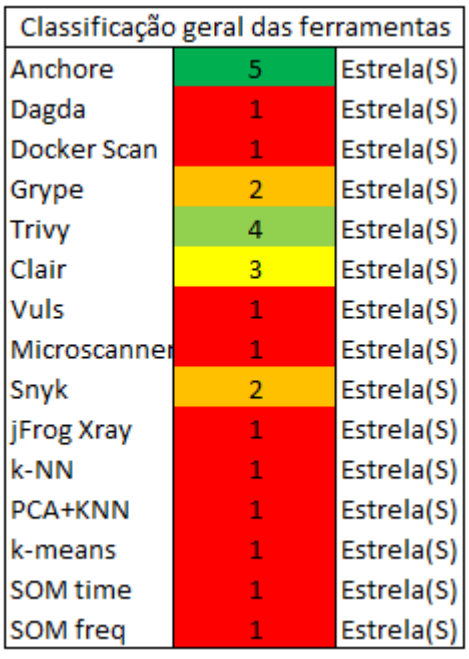


Figura 1. Resultados da aplicação do método AHP nos resultados de todos os estudos.

A análise dos resultados revela que a ferramenta Anchore se destacou significativamente, obtendo a pontuação máxima (5 estrelas) na classificação geral. Esse

[2] https://github.com/ali-id/AHP-Calc-Worksheets

desempenho notável é confirmado pela consistência das avaliações elevadas que a Anchore recebeu em diversos estudos, como evidenciado na Figura 2. Especificamente, a Anchore foi objeto de análise em 7 estudos diferentes, demonstrando um desempenho relevante, isto é: obteve a nota máxima (5 estrelas) em 4 estudos, recebeu 4 estrelas em dois estudos e alcançou 3 estrelas em um dos estudos. Essa consistência nas avaliações, juntamente com a ampla cobertura em diferentes estudos, contribuiu significativamente para a classificação de topo da Anchore na avaliação geral.

A ferramenta Trivy demonstrou um desempenho notável, ocupando a segunda posição na classificação geral, como ilustrado na Figura 1. Essa posição é resultado de um desempenho consistente nos estudos em que foi avaliada. Especificamente, o Trivy foi objeto de análise em 4 estudos diferentes e, em cada um deles, alcançou a classificação máxima de 5 estrelas, como pode ser observado na Figura 2. Essa consistência na obtenção da pontuação máxima é um indicador significativo da eficácia e confiabilidade da ferramenta Trivy na detecção de vulnerabilidades em imagens Docker.

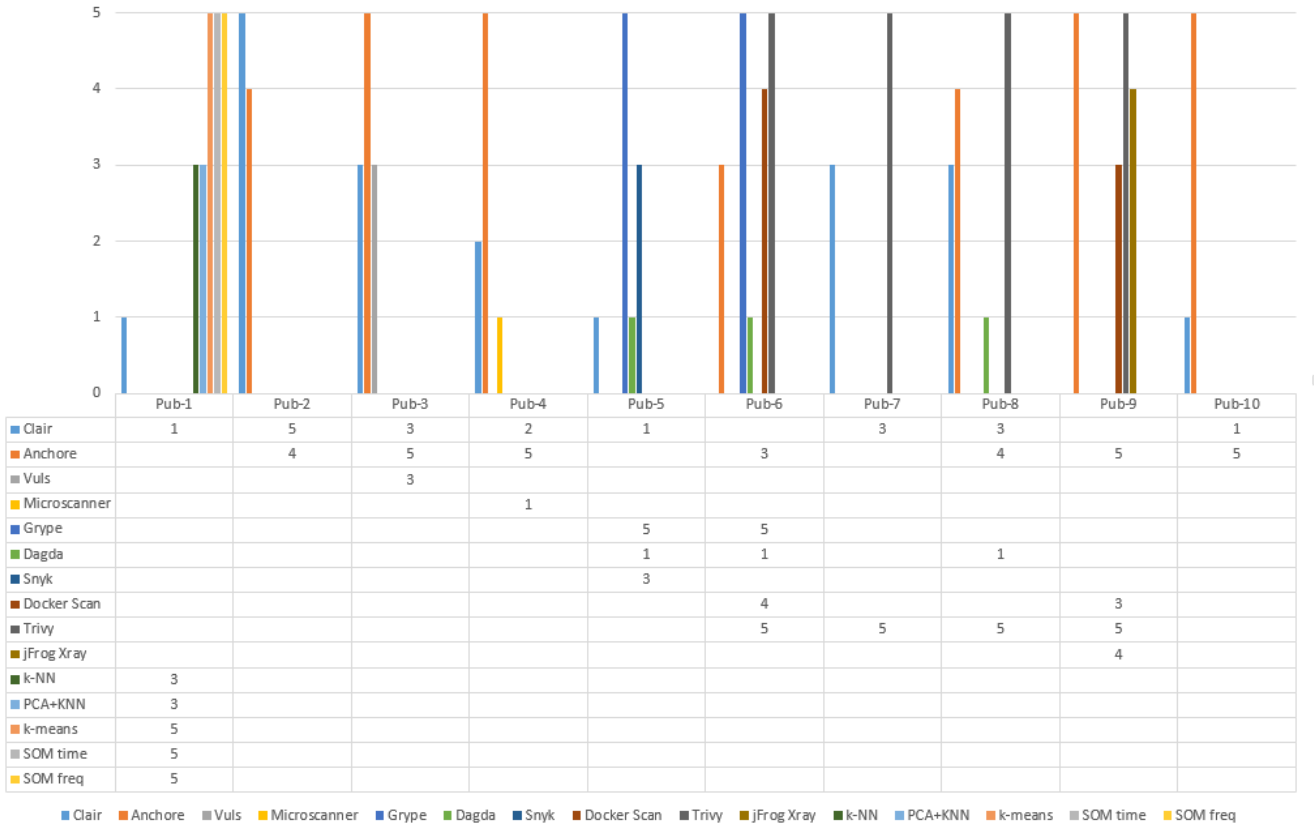
Um aspecto particularmente interessante dos resultados é a comparação direta entre Trivy e Anchore nos estudos em que ambas foram avaliadas conjuntamente. Especificamente, nos estudos identificados como Pub-6 e Pub-8, o Trivy superou o desempenho da Anchore. Esse resultado sugere que, em determinados contextos ou para certos tipos de análise, o Trivy pode oferecer vantagens sobre a Anchore. O fato de o Trivy superar a Anchore em comparações diretas, mesmo ocupando a segunda posição geral, destaca a importância de considerar não apenas as classificações gerais, mas também os resultados de comparações específicas ao escolher uma ferramenta para um determinado ambiente ou caso de uso.

A ferramenta Clair ocupou a terceira posição na classificação geral, conforme ilustrado na Figura 1. No entanto, uma análise mais detalhada de seu desempenho nos estudos individuais revela um padrão de resultados bastante variado e inconsistente. Ao examinar a Figura 2, que apresenta os resultados da aplicação do método AHP por ferramenta e por estudo, observa-se uma significativa flutuação no desempenho da Clair, isto é: em um dos estudos, a Clair alcançou a classificação máxima de 5 estrelas, demonstrando excelente eficácia na detecção de vulnerabilidades. Por outro lado, em três estudos diferentes, a Clair recebeu a classificação mínima de 1 estrela, indicando um desempenho consideravelmente abaixo das expectativas nesses cenários. Nos demais estudos, o desempenho da Clair variou entre esses extremos. A inconsistência observada no desempenho da Clair é um aspecto importante a ser considerado por profissionais e pesquisadores ao avaliar essa ferramenta para uso em seus projetos.

A ferramenta Grype, desenvolvida pela empresa Anchore, demonstrou um desempenho notável nos estudos em que foi avaliada. Conforme ilustrado na Figura 2, a Grype alcançou a classificação máxima de 5 estrelas nos dois estudos que a analisaram. Esse resultado é particularmente relevante, considerando que a Grype é uma solução relativamente nova no mercado de ferramentas de análise de vulnerabilidades em imagens Docker. Apesar de sua classificação geral ter sido impactada pelo número limitado de estudos que a avaliaram, a Grype mostrou-se altamente competitiva. Em diversas métricas de avaliação, ela superou ferramentas mais testadas como Anchore e Trivy. Esse desempenho superior em comparações diretas sugere que a Grype apresenta potencial para a detecção de vulnerabilidades em contêineres. Vale ressaltar que em janeiro de 2023 a Anchore, em sua versão de código aberto, foi oficialmente descontinuada [23]. Como substituta, a empresa desenvolvedora recomenda o uso das ferramentas Grype e Syft, ou a versão comercial Anchore Enterprise.

Essa transição da Anchore para a Grype como a principal ferramenta recomendada pela empresa é um fator importante a ser considerado. Ela sugere que a Grype pode incorporar melhorias e avanços baseados na experiência acumulada com o desenvolvimento e uso da Anchore e que o foco de desenvolvimento e suporte da empresa está agora direcionado para a Grype, o que pode resultar em atualizações e melhorias mais frequentes. O excelente desempenho da Grype nos estudos disponíveis, combinado com sua posição como sucessora recomendada da Anchore, sugere que essa ferramenta merece atenção especial de profissionais e pesquisadores da área de segurança de contêineres. Embora mais estudos sejam necessários para estabelecer uma classificação geral mais robusta, os resultados apresentados indicam que a Grype pode se tornar uma das principais ferramentas no campo de análise de vulnerabilidades em imagens Docker.

As ferramentas Docker Scan e JFrog Xray apresentaram resultados intermediários em suas avaliações individuais, oferecendo *insights* interessantes sobre seu desempenho na análise de vulnerabilidades em imagens Docker. O JFrog Xray foi avaliado em apenas um estudo, no qual obteve uma classificação de 4 estrelas. Esse resultado indica um desempenho satisfatório, sugerindo que a ferramenta é capaz de realizar análises



	Pub-1	Pub-2	Pub-3	Pub-4	Pub-5	Pub-6	Pub-7	Pub-8	Pub-9	Pub-10
Clair	1	5	3	2	1		3	3		1
Anchore		4	5	5		3		4	5	5
Vuls			3							
Microscanner				1						
Grype					5	5				
Dagda					1	1		1		
Snyk					3					
Docker Scan						4			3	
Trivy						5	5	5	5	
jFrog Xray									4	
k-NN	3									
PCA+KNN	3									
k-means	5									
SOM time	5									
SOM freq	5									

Figura 2. Resultados da aplicação do método AHP por ferramenta e por estudo.

de vulnerabilidades com potencial.

Por outro lado, o Docker Scan apresentou resultados mais variados nos estudos em que foi analisado. Em um estudo alcançou a classificação máxima de 5 estrelas, demonstrando excelente desempenho, e 3 estrelas em outro, indicando um desempenho moderado. Essa variação nos resultados do Docker Scan sugere que sua eficácia pode depender do contexto específico ou dos tipos de vulnerabilidades analisadas em cada estudo. Um ponto importante a ser destacado em relação ao Docker Scan é a observação feita no estudo Pub-6. Nesse estudo, a métrica de número de vulnerabilidades identificadas foi excluída da aplicação do método AHP para o Docker Scan. O motivo dessa exclusão foi a constatação de que 77% das vulnerabilidades detectadas por essa ferramenta eram duplicadas. Essa informação indica uma potencial limitação do Docker Scan em distinguir entre vulnerabilidades únicas e duplicadas, e o número bruto de vulnerabilidades detectadas pode não ser um indicador confiável da eficácia da ferramenta. A exclusão dessa métrica no estudo Pub-6 representa uma abordagem metodológica importante, pois evita que o desempenho do Docker Scan seja artificialmente inflado por detecções duplicadas.

A ferramenta Snyk, embora avaliada em apenas um estudo, demonstrou um desempenho mediano. Nesse único estudo em que foi analisada, a Snyk alcançou a segunda colocação, ficando atrás apenas da ferramenta Grype. Nesse estudo, a Snyk superou outras ferramentas bem estabelecidas como Clair e Dagda em termos de eficácia na detecção de vulnerabilidades. Esse desempenho da Snyk sugere que, apesar da limitada avaliação, a ferramenta possui potencial significativo para a análise de vulnerabilidades em imagens Docker. No entanto, é importante ressaltar que uma avaliação mais abrangente, envolvendo múltiplos estudos e diferentes cenários, seria necessária para confirmar a consistência desse desempenho.

Em contraste, outras ferramentas de análise estática, incluindo Vuls, Dagda e MicroScanner, apresentaram resultados menos expressivos em comparação com as ferramentas líderes. Essas ferramentas obtiveram classificações inferiores tanto nos resultados individuais de cada estudo quanto na classificação geral consolidada. Vuls demonstrou um desempenho abaixo da média nos estudos em que foi avaliada, indicando possíveis limitações em sua capacidade de detecção de vulnerabilidades em comparação com outras ferramen-

tas. A ferramenta Dagda, apesar de ser avaliada em múltiplos estudos, consistentemente recebeu classificações baixas, sugerindo que pode ter dificuldades em competir com ferramentas mais avançadas em termos de eficácia de detecção. Por fim, MicroScanner apresentou resultados menos expressivos, indicando que pode não oferecer o mesmo nível de detecção de vulnerabilidades que as ferramentas líderes do mercado.

Na análise das ferramentas de detecção dinâmica de vulnerabilidades em imagens Docker, os algoritmos SOM (*Self-Organizing Map*) e K-means demonstraram um desempenho excepcional em suas avaliações individuais, ambos alcançando a pontuação máxima de 5 estrelas. Entre esses dois, o algoritmo SOM apresentou uma ligeira vantagem sobre o K-means, indicando uma eficácia marginalmente superior na detecção de vulnerabilidades. Em contraste, os algoritmos K-NN (*K-Nearest Neighbors*) e PCA+KNN (*Principal Component Analysis + K-Nearest Neighbors*) apresentaram um desempenho moderado, recebendo uma classificação de 3 estrelas. Nessa comparação, o PCA+KNN demonstrou uma pequena vantagem, sugerindo que a adição da análise de componentes principais (PCA) oferece uma melhoria, ainda que modesta, na eficácia da detecção. Esses resultados das avaliações individuais sugerem que os algoritmos SOM e K-means podem ser particularmente eficazes na análise dinâmica de vulnerabilidades em imagens Docker, superando significativamente as abordagens baseadas em K-NN. No entanto, um aspecto importante a ser considerado é o resultado da classificação geral, que levou em conta todos os estudos analisados na revisão sistemática. Nessa classificação geral, todas as ferramentas de análise dinâmica obtiveram apenas 1 estrela. Esse resultado, aparentemente contraditório em relação às avaliações individuais, indica uma clara necessidade de conduzir mais estudos comparativos envolvendo essas ferramentas de análise dinâmica e outras ferramentas de análise de vulnerabilidades.

4 Conclusão

Com base nos resultados apresentados neste trabalho, conclui-se que as ferramentas Anchore e Trivy se destacaram consistentemente como as mais eficazes e precisas na detecção de vulnerabilidades em imagens Docker. Seu desempenho superior em múltiplos estudos as posiciona como opções confiáveis para profissionais e pesquisadores da área. Embora avaliada em apenas dois estudos, a ferramenta Grype demonstrou um alto desempenho, indicando seu potencial como uma solução eficaz. Seu desenvolvimento como sucessora da Anchore sugere que ela pode se tornar uma das principais ferramentas no futuro próximo. A ferramenta Clair apresentou resultados inconsistentes, com desempenho insatisfatório na maioria dos estudos. Essa variabilidade ressalta a importância de considerar o contexto específico de uso ao selecionar ferramentas de análise de vulnerabilidades. Para as demais ferramentas de análise estática, a revisão sistemática evidenciou uma clara necessidade de mais estudos avaliativos. Essa lacuna na pesquisa representa uma oportunidade importante para futuros trabalhos na área. Outro ponto importante a ser destacado é que a utilização combinada de diferentes ferramentas pode resultar em uma detecção mais abrangente e precisa de vulnerabilidades. Essa abordagem multiferramenta pode compensar as limitações individuais de cada solução. No campo da análise dinâmica, os algoritmos SOM e K-means apresentaram os melhores resultados. No entanto, a escassez de estudos comparativos entre essas abordagens dinâmicas e as ferramentas de análise estática indica outra importante área para pesquisas futuras.

Referências

1 Ishizaka, A.; Lusti, M. How to derive priorities in AHP: A comparative study. *Central European Journal of Operations Research*, v. 14, 2006. DOI: 10.1007/s10100-006-0012-9. Disponível em: https://link.springer.com/article/10.1007/s10100-006-0012-9. Acesso em: 28 mai. 2024.

2 Shu, R.; Gu, X.; Enck, W. A Study of Security Vulnerabilities on Docker Hub. *In*. DOI: 10.1145/3029806.3029832. Disponível em: https://dl.acm.org/doi/10.1145/3029806.3029832. Acesso em: 25 ago. 2024.

3 Souppaya, M.; Morello, J.; Scarfone, K. *Application Container Security Guide*. 2017. DOI: 10.6028/NIST.SP.800-190. Disponível em: https://csrc.nist.gov/pubs/sp/800/190/final. Acesso em: 23 jun. 2024.

4 Alyas, T. *et al.* Container Performance and Vulnerability Management for Container Security Using Docker Engine. *Security and Communication Networks*, v. 2022, 2022. DOI: https://doi.org/10.1155/2022/6819002. Disponível em: https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/6819002. Acesso em: 15 ago. 2024.

5 Martin, A. *et al.* Docker ecosystem – Vulnerability Analysis. *Computer Communications*, v. 122, 2018. DOI: https://doi.org/10.1016/j.comcom.2018.03.011. Disponível em: https://www.sciencedirect.com/science/article/pii/S0140366417300956. Acesso em: 8 ago. 2024.

6 Liu, P. *et al.* Understanding the Security Risks of Docker Hub. *In*: COMPUTER Security – ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I. Springer-Verlag, 2020. DOI: 10.1007/978-3-030-58951-6_13. Disponível em: https://dl.acm.org/doi/10.1007/978-3-030-58951-6_13. Acesso em: 14 jun. 2024.

7 Brady, K. *et al.* Docker Container Security in Cloud Computing. *In*: 2020 10th Annual Computing and Communication Workshop and Conference (CCWC). 2020. P. 975–980. DOI: 10.1109/CCWC47524.2020.9031195. Disponível em: https://ieeexplore.ieee.org/document/9031195. Acesso em: 3 ago. 2024.

8 Tunde-Onadele, O. *et al.* A Study on Container Vulnerability Exploit Detection. *In*: 2019 IEEE International Conference on Cloud Engineering (IC2E). 2019. P. 121–127. DOI: 10.1109/IC2E.2019.00026. Disponível em: https://ieeexplore.ieee.org/document/8790061. Acesso em: 27 jun. 2024.

9 Pinnamaneni, J.; S, N.; Honnavalli, P. Identifying Vulnerabilities in Docker Image Code using ML Techniques. *In*: 2022 2nd Asian Conference on Innovation in Technology (ASIANCON). 2022. P. 1–5. DOI: 10.1109/ASIANCON55314.2022.9908676. Disponível em: https://ieeexplore.ieee.org/document/9908676. Acesso em: 7 ago. 2024.

10 JAGELID, M. *Container Vulnerability Scanners: An Analysis*. 2020. MASTER'S THESIS – KTH Royal Institute of Technology, School of Electrical Engineering e Computer Science. Disponível em: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1463231&dswid=-9391. Acesso em: 30 jun. 2024.

11 Kaur, B. *et al.* An analysis of security vulnerabilities in container images for scientific data analysis. *GigaScience*, v. 10, 2021. Disponível em: https://academic.oup.com/gigascience/article/10/6/giab025/6291571. Acesso em: 11 jun. 2024.

12 Javed, O.; Salman, T. An Evaluation of Container Security Vulnerability Detection Tools. *In*: 2021 5th International Conference on Cloud and Big Data Computing. 2021. Disponível em: https://uslc-lab.github.io/assets/papers/javedandtoor_CBDC2021.pdf. Acesso em: 18 ago. 2024.

13 Fialho, Y.; Bordim, J. Segurança em imagens Docker: um estudo de ferramentas de análise estática. *In*: ANAIS do XXVI Workshop de Gerência e Operação de Redes e Serviços. SBC, 2021. P. 138–151. DOI: 10.5753/wgrs.2021.17191. Disponível em: https://sol.sbc.org.br/index.php/wgrs/article/view/17191.

14 Andersson, M.; Berg, R. H. *Docker Container Images: Concerns about available container image scanning tools and image security*. 2022. Disponível em: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1679447&dswid=-3705. Acesso em: 1 ago. 2024.

15 Zarei, M. *Investigating the inner workings of container image vulnerability scanners*. 2022. MASTER'S THESIS – Oslo Metropolitan University, Faculty of Technology, Art e Design. Disponível em: https://oda.oslomet.no/oda-xmlui/handle/11250/3017416. Acesso em: 13 jul. 2024.

16 Pihlak, A. *CONTINUOUS DOCKER IMAGE ANALYSIS AND INTRUSION DETECTION BASED ON OPEN-SOURCE TOOLS*. 2020. MASTER'S THESIS – TALLINN UNIVERSITY OF TECHNOLOGY, School of Information Technology. Disponível em: https://digikogu.taltech.ee/en/Item/480851c3-18ff-44f2-b0f5-22db44f44dd5. Acesso em: 1 jul. 2024.

17 Malhotra, R.; Bansal, A.; Kessentini, M. Vulnerability Analysis of Docker Hub Official Images and Verified Images. *In*: 2023 IEEE International Conference on Service-Oriented System Engineering (SOSE). 2023. DOI: 10.1109/SOSE58276.2023.00025. Disponível em: https://ieeexplore.ieee.org/document/10254755. Acesso em: 1 ago. 2024.

18 Bhardwaj, P. *Detecting Container vulnerabilities leveraging the CICD pipeline*. 2023. Diss. (Mestrado) – National College of Ireland, School of Computing. Disponível em: https://norma.ncirl.ie/6512/. Acesso em: 20 jun. 2024.

19 Triantaphyllou, E. *Multi-Criteria Decision Making Methods: A Comparative Study*. Springer, 2000. v. 44. ISBN 978-1-4419-4838-0. DOI: 10.1007/978-1-4757-3157-6.

20 Bernasconi, M.; Choirat, C.; Seri, R. The Analytic Hierarchy Process and the Theory of Measurement. *University of Venice "Ca' Foscari", Department of Economics, Working Papers*, v. 56, 2009. DOI: 10.2307/27784145. Disponível em: https://www.dse.univr.it/documenti/Seminario/documenti/documenti803241.pdf. Acesso em: 19 ago. 2024.

21 Badri, M. Combining the analytic hierarchy process and goal programming for global facility location-allocation problem. *International Journal of Production Economics*, v. 62, 1999. DOI: https://doi.org/10.1016/S0925-5273(98)00249-7. Disponível em: https://www.sciencedirect.com/science/article/pii/S0925527398002497. Acesso em: 8 jul. 2024.

22 Vargas, L. G. An overview of the analytic hierarchy process and its applications. *European Journal of Operational Research*, v. 48, 1990. DOI: https://doi.org/10.1016/0377-2217(90)90056-H. Disponível em: https://www.sciencedirect.com/science/article/pii/037722179090056H. Acesso em: 10 jul. 2024.

23 Anchore. *Anchore Engine*. Anchore, Inc, 2023. Disponível em: https://github.com/anchore/anchore-engine. Acesso em: 6 jun. 2024.