Computers & Security 135 (2023) 103478

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Longitudinal risk-based security assessment of docker software container images

Alan Mills *, Jonathan White, Phil Legg

Computer Science Research Centre, University of the West of England, Bristol, UK

ARTICLE INFO

Keywords:
Container security
Vulnerability analysis
CVE

ABSTRACT

As the use of software containerisation has increased, so too has the need for security research on their usage, with various surveys and studies conducted to assess the overall security posture of software container images. To date, there has been very little work that has taken a longitudinal view of container security to observe whether vulnerabilities are being resolved over time, as well as understanding the real-world implications of reported vulnerabilities, to assess the evolving security posture. In this work, we study the evolution of 380 software container images across 3 analysis periods between July 2022 and January 2023 to analyse maintenance and vulnerabilities factors over time. We sample across the 3 DockerHub categories: Official, Verified and OSS (Sponsored) Open Source Software. We found that the number of vulnerabilities present increased over time despite many containers receiving regular updates by providers. We also found that the choice of container OS can dramatically impact the number of reported vulnerabilities present over time, with Debian-based images typically having many more vulnerabilities that other Linux distributions, and with some containers still reporting vulnerabilities that date back as far as 1999. However, when taking into account additional reported attributes such as the attack vector required and the existence of a public exploit rated higher than negligible, we found that for each analysis period, less than 1% of all vulnerabilities present what we would consider as high risk real-world impact. Through our investigation, we aim to improve the understanding of the threat landscape posed by software containerisation that is further complicated by the discrepancies between different vulnerability reporting tools.

1. Introduction

Software containerisation is fundamental to the modern computing paradigms of Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS), by providing a lightweight virtualised workspace without the requirements of a full virtual machine provision. Docker has gained wide spread adoption of its online platform, DockerHub (2023), that provides a library of pre-built containers that can be downloaded and deployed easily. As per the containerisation analogy, all necessary components are pre-packaged and available within the container, making for simplified management by software teams. Whilst this provides flexibility and convenience to software development teams, the increased adoption and reliance of pre-built containers also gives rise to the growing concerns of software supply chain security, be it deliberate or unintentional that vulnerable or malicious software may become invoked. Cases such as the Solarwinds Orion platform (CSO, 2020) and the PyTorch library (CSO, 2023) have drawn much media attention that has highlighted the increasing issues related to software supply chain security.

As containerisation continues to grow as a modern software development paradigm, it is vital to understand the context of the security implications highlighted by existing scanning tools. There are a wealth of existing container scanning tools, including Clair (2023), Docker Scan (2023), Grype (2023), JFrog and Docker Security (2023), Sysdig secure (2023) and Trivy (2023), that are designed to identify and grade vulnerabilities between negligible, low, medium, high, and critical. In previous work (Mills et al., 2022), we have observed significant discrepancies in the reporting across these tools, such as that of CVE-2019-3843 and CVE-2019-3844, for Set User ID and Set Group ID vulnerabilities. We previously proposed a tool, OGMA, that enables a unified analysis across these multiple scanning tools, whilst also factoring in severity, patch status and existence of an exploit, to assess the overall potential impact and real-world risk of reported vulnerabilities. In the case of these two CVEs, both vulnerabilities require local access in order for

* Corresponding author.
E-mail address: alan.mills@uwe.ac.uk (A. Mills).

https://doi.org/10.1016/j.cose.2023.103478
Received 16 May 2023; Received in revised form 22 August 2023; Accepted 7 September 2023
Available online 14 September 2023
0167-4048/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

the published exploit to be successful, and within a containerised environment, it is quite often that the default user is root in any case. Whilst contextual details are often overlooked in current scanning tools, OGMA will identify these issues, which actually render the CVE as irrelevant within the setting of a container environment.

In this study we conduct, what is to the best of our knowledge, the first longitudinal security analysis on containerisation security. We study 380 software container images across 3 analysis periods between July 2022 and January 2023 to analyse maintenance and vulnerabilities, and to observe how the number of vulnerabilities may increase or decrease over time, as both new functionality is introduced and as security vulnerabilities are identified, reported and fixed. We use OGMA (Mills et al., 2022) to provide unified results from 6 different scanners, to assess severity, attack vector, patch status and the existence of exploits (within Exploit-DB (2023)). The sample images are drawn from 3 categories: Official, Verified and (Sponsored) Open Source Software (OSS). We found that the number of vulnerabilities present increased over time despite many containers receiving regular updates by providers. We also found that the container OS can dramatically impact the number of reported vulnerabilities present over time, with Debian-based images typically having many more vulnerabilities that other Linux distributions, and with some containers still reporting vulnerabilities that date back as far as 1999. We also found that 26%-33% of CVEs were either local vulnerabilities or had been designated as “NOFIX” issues by the package maintainers. Across each analysis interval, we found that the number of CVEs with exploits was 3% of the total number of reported CVEs, dropping to 1% when we removed exploits for local CVEs and less than 1% when we further removed those categorised as negligible. Our analysis also showed that OSS images had fewer vulnerabilities than Official or Verified across all analysis periods, and highlight the importance of security-based update management, with example of images being updated per analysis run still showing an increase in vulnerabilities and exploits.

Our study shows the evolving state of containerisation security across a large sample of images and highlights the importance of context-based vulnerability analysis, a switch from previous surveys and studies which have focused on both single time periods and severity or age as metrics for assessing the state of container security. Our findings show that while significant numbers of vulnerabilities are often reported using these metrics, a far smaller number, the highest number in a single analysis period being 390 or less than 1% of all vulnerabilities, present what we would consider a high risk, potentially real-world impact. We aim to increase awareness for security researchers and software development teams by shifting focus to these smaller numbers of vulnerabilities and promote best practice of scanning and consideration of the real-world security implications where using third party libraries and tools in workflows.

2. Related works

For our related works, we report on previous research that has examined container-based security assessment. We highlight the key details of previous works as a means to guide further research, notably where repeat analysis has not been performed to observe whether security issues improve or degrade over time.

Sultan et al. (2019) provide a survey on container security, highlighting the key issues, challenges and future directions. They consider six main forms of risk: image risks, registry risks, orchestrator risks, container risks, host OS risks, and other risks. Furthermore, they present four generalised use cases to assess how previous works have addressed the challenges that prevail in container-based security. The use cases are defined as: protecting a container from applications inside it, inter-container protection, protecting the host from containers, and protecting containers from a malicious or semi-honest host. This research provides a comprehensive overview of the container security landscape, which has since provoked the development of container scanning and analysis tools that are the primary focus of our research.

A 2016 study by Shu et al. (2017) presented the DIVA (Docker Image Vulnerability Analysis) framework, which they used to analysis over 350,000 container images. They found on average 180 vulnerabilities across the official and community images tested, with over 80% of these images having at least 1 high severity vulnerability. It is noted however that their framework incorporates only a single scanner (Clair), where other scanner tools may have provided different results. In addition, whilst they do investigate image age as a factor of container security, they only perform container analysis for a single time step and do not consider how the security posture of the container images may vary over time. Zerouali et al. (2019) focus their analysis on the concept of “technical lag”, looking at the correlation between container updates, outdated packages and vulnerabilities for over 7000 Debian based images from the Official and Community categories. Using the Debian Security Bug Tracker they found that time to fix ranged from 2.1 months for high severity vulnerabilities through to 27 months for 50% of low severity vulnerabilities, however they noted that Community images were on average more up to date than Official images. Their analysis showed that while all containers were impacted by a vulnerability of some kind, these could be traced back to only a small portion (12.2%) of unique installed packages, with a fixed version listed for almost 50% of the reported vulnerabilities. However their study included a significant proportion (48% of Community and 66%) of images using older Debian releases that had not been regularly updated and the authors acknowledge that some vulnerabilities were not reported due to missing debianbug id’s. In later work by Liu et al. (2020), they study CVEs within docker images, as well as the prevalence of malicious images and sensitive parameters. They focus on vulnerability age using Anchore to detect and identify vulnerabilities and reported on delays in container patch management. They found an average delay of 422 days for patches to be applied to images, and they discuss this in the context of the time available to an attacker to therefore exploit vulnerabilities. It is noted that the details around the vulnerabilities and the potential impact of exploitation are not discussed in this work, despite the significance that this would have on the likelihood and severity of risk. Wist et al. (2021) expand on an earlier thesis where they analyse 2500 docker images from Official, Verified, Certified and Community images. They compare the vulnerability findings for images within each category, with a focus on vulnerability severity and age. Whilst Official images were deemed to be the most secure (in comparison to Community, Verified and Certified images), they still found 45.9% of Official images had one or more critical or high severity CVE, and only 17.8% of the images analysed contained no CVEs at all. Whilst they only state that they use an “open source vulnerability scanner”, it is inferred from their GitHub repository that this was also based on Anchore. As with other previous works, the results could well appear different with an alternative scanning tool being deployed. Haque and Babar (2022) highlight the lack of research that looks at the impact and exploitability of container vulnerabilities. In their work they analysed 261 Official base images (which are themselves built on to create container images) using Anchore. They identified 1,983 unique vulnerabilities and used Exploit-DB to map 74 exploits to vulnerabilities. Their results highlighted the breakdown of impact and exploitability within reported vulnerabilities, but included local privilege escalation vulnerabilities as HI (High Impact), potentially not taking into account container use case, such as a default root user, when assessing vulnerability impact. As their analysis was focused on base images from a single category there is scope to build on this work by including multiple image categories, as well as analysis of parent and child images to further evaluate vulnerability inheritance within container images.

Javed and Toor (2021) analyse 59 container images using three open source container scanners (Clair, Anchore and Microscanner). Their results highlight the differences in coverage and accuracy between the tools analysed and advocate a combination of static and dynamic anal-

ysis. Similarly in our previous research, Mills et al. (2022) addressed visualising the difference between six container scanning tools to provide more-informed analysis, where we scanned 25 popular containers to reveal the reported discrepancies between the common and popular scanning tools. Whilst both of these works do account for the variation between scanners, they both only study a smaller number of container images, and they also do not look at container security over time as we address in this current study. Chen et al. (2022) present SEAF (A Scalable, Efficient, and Application-independent Framework for container security detection) that uses differential analysis of docker image layers to quickly identify where a docker image layer has been altered, reducing duplicate analysis of the same layers across multiple images. The authors used SEAF to collect metadata for over 12 million images, however they present results for only a small set of case studies with limited detail around the vulnerabilities themselves. The authors utilise two different scanners, Clair and Dagda, for comparative analysis. It is noted that Dagda has not been updated for almost 2 years (as of May 2023) and has multiple open issues potentially impacting it's use case (Dagda, 2021). Anchore, which has also been used by multiple previous studies, also became unsupported as of 2023, replaced in part by Grype (Anchore-engine, 2023), further emphasising the need for multiple scanning tools to gain a comparative analysis, where some open-source tools may simply not be up to date or well maintained.

Finally, Jacobs *et al.* discuss the prevalence of reported vulnerabilities and those that are actually exploited (Jacobs et al., 2020). They look at trends and modelling which can be used to identify vulnerabilities which are more likely to be exploited, rather than relying on the existing system of vulnerability severity. While the CVSS score has some impact on the likelihood that a vulnerability will be exploited, rising from 1.3% for a score of 2 through to 18.4% for a score of 9, they also identify key considerations such as the vendor of the impacted software package, attack type and existence of an exploit that impact the likelihood of a vulnerability being exploited. This highlights both the relatively small number of vulnerabilities which are exploited as well as the need to frame our consideration and concern beyond just the vulnerability severity or age. As a result, they go on to propose a Exploit Prediction Scoring System (EPSS) (Jacobs et al., 2021) to address the shortcomings of the CVSS scheme. We apply similar principals to our own analysis of container image data, such that we frame the scanner results in the context of whether vulnerabilities are likely to manifest as having a real-world impact.

Whilst the studies discussed above are indeed valuable, they tend to only provide a single snapshot of the state of container image security, at the time of writing, and do not consider the evolution of containers as live software deployments that may receive updates over time. The analysis of vulnerabilities tends to address only severity and age (exceeding 400 days in some cases (Liu et al., 2020)), where discussion on issues of patch status, attack vectors and context of vulnerabilities is typically limited or missing. Likewise, previous works tend to rely on a single scanning tool such as Clair or Anchore, however both the authors of Javed and Toor (2021) and our previous work Mills et al. (2022) have demonstrated the variability of results from multiple scanners and risk of deprecation for open-source tooling, and therefore the importance of taking a unified result from multiple scanners. These issues highlight the shortcomings of these studies being indicative of the state of container security for a given year (as stated in Wist et al. (2021)). We therefore believe that our longitudinal analysis over a substantial period of time, accounting for container image vulnerabilities beyond just age and severity, is required to provide a real-world, context-driven analysis of current software containerisation security.

3. Methodology

To conduct a realistic and representative analysis of publicly-available container images, we sampled across the 3 image categories; Official, Verified and (Sponsored) Open Source Software. Our sample originally consisted of over 400 images, with at least 100 from each of the 3 images categories stated. Image selection was initially based on the number of pulls, also accounting for the repository to avoid a single category being comprised entirely of images from one maintainer. Over the period of our analysis, some images became deprecated, causing the total number of images to be analysed over the full time period to be 380. We purposefully did not include images which have been listed as deprecated or those that appeared to have been created for a single use function, and are therefore no longer actively used or maintained, such as hola-mundo.[1] These guiding principles for conducting image selection ensured that our sample was relevant and representative of popular containers that have significant user bases, whilst also ensuring that it is fair, so that containers that are no longer supported do not unfairly skew the resulting analysis.

We treat each of the three analysis stages as separate events, where each analysis stage is completed within a time-bounded 30-day period. This was to minimise the impact that vulnerability discovery and disclosure would have on the images analysed. To further mitigate against potential changes that may occur during each analysis stage, all images were analysed in tandem by six independent scanners using the OGMA tool (Mills et al., 2022). The six independent scanning tools used are Clair (2023), Docker Scan (2023), Grype (2023), JFrog and Docker Security (2023), Sysdig secure (2023) and Trivy (2023). Images were downloaded and added to separate DockerHub repositories (one per analysis period) to ensure that an exact clone could be retrieved at a later date if required for the purposes of this research, and to ensure that no external changes had occurred in the software source code. The three analysis stages were completed in July 2022, October 2022, and January 2023.

Following the initial analysis period, subsequent periods would utilise the latest available version of the same image. In some cases the image tag would be noticeably different, indicating version changes, such as the official Elasticsearch which changed from 8.2.2 to 8.4.2. In other instances the image itself may have been updated but the tag would remain the same, such as for the official adminer image which remained at version (and tag) 4.8.1. Where multiple version changes had occurred, we chose to use the same major software version, unless it appeared to have been deprecated, for example, where no changes had been made to the supposed "current" major version within 2 months, but where a new major version had also been released more recently. We provide full details of the images analysed and their update status during the full analysis period in Appendix A.

4. Analysis

Following the scanning of each container image, the OGMA platform (Mills et al., 2022) was also used to highlight vulnerability severity, attack vector, patch status and the existence of an exploit on Exploit-DB (2023). Results from the multiple scanners were resolved to the *highest* reported severity, so where two scanners may report medium, if another scanner reported the vulnerability as high then the vulnerability would be categorised overall as high, so as to capture the most significant rating across all scanners. This decision was made based on previously identified discrepancies with vulnerability reporting across different scanners (Mills et al., 2022). For example the CVE-2019-9883 is reported as either "negligible" (Clair and Grype) or "low" (Docker Scan and Trivy), however on NVD this has a CVSS3 score of 8.8 (high).[2] It is also worth noting that despite reporting this CVE as "low" Trivy also includes a CVSS3 score of 9.8 for the CVE itself. Similar issues can be found in other CVEs such as CVE-2021-22945[3] and CVE-2022-24303,[4]

[1] https://hub.docker.com/_/hola-mundo.
[2] https://nvd.nist.gov/vuln/detail/CVE-2019-9883.
[3] https://nvd.nist.gov/vuln/detail/CVE-2021-22945.
[4] https://nvd.nist.gov/vuln/detail/CVE-2022-24303.

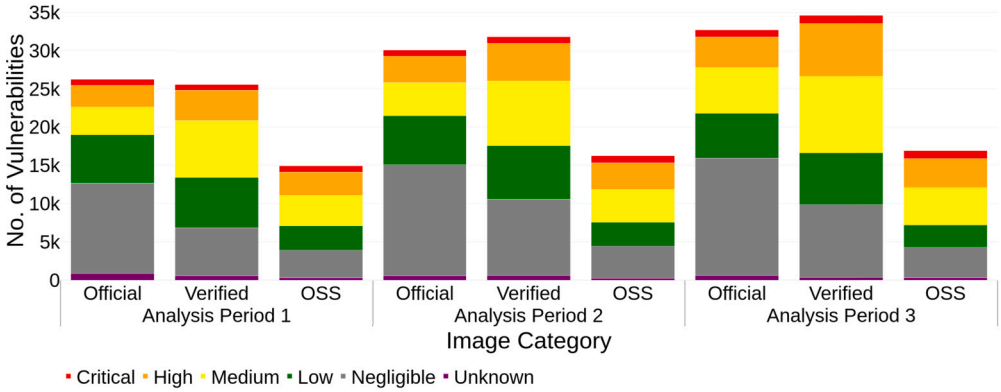


Fig. 1. Results to show the number of vulnerabilities by image category (Official, Verified, OSS) for each analysis period (excluding NOFIX entries).

both of which have reporting ranging from "negligible" to "critical" dependent on the scanner. These disparities amongst CVE reporting not only between scanners, but within the scanner results themselves makes reporting based on either a mean or highest voted severity potentially misleading. In the case of CVE-2021-22945 for example this could be reported as "negligible" (based on a common output from 2 scanners), "critical" (based on the same) or "medium" (based on a mean severity – but not reported by any scanner). By opting for the highest reported severity we set a simple unified output metric which covers the most significant reporting and is accurate to at least 1 instance of scanner results in all situations. Further analysis of the reporting discrepancy is also available in our supplementary materials, however is omitted here in the interest of brevity. This is available at our GitHub project repository: https://github.com/uwe-cyber/Longitudinal-Container-Study-2023.

Use of the OGMA platform also allowed us to easily identify vulnerabilities which had been labelled as "NOFIX". These are vulnerabilities that have no available, or intended patch to be released. This can occur where there is disagreement of whether a reported vulnerability is actually a security threat, for example, if the conditions required to exploit are too specific or outside the intended use case, or if the impacted product is no longer supported.

To further enrich the data and provide additional context for vulnerabilities, they were mapped to the associated CWE (Common Weakness Enumeration) ID and the default user for each scanned image extracted. For example, in the case of a local vulnerability which impacts privilege management, where a containers default user is root this vulnerability would have no real-world impact. For a malicious actor to exploit the vulnerability they would need to have local or shell access to the container environment, not as the default user. This would likely require them to either already have access to the container instance or to have exploited a remote vulnerability within the container, which itself would need to allow them the ability to execute code or otherwise establish command execution. In the latter scenario this initial vulnerability would have already been identified as a higher risk.

5. Results

For our results, we analyse the number of vulnerabilities for different groupings of the container images studied over the 3 analysis periods, as well as the number of CVEs present in the container images which have publicly-available exploits. We also investigate the number of CVEs in relation to the size of the container images. We further investigate the role that CVE inheritance plays within images that share the same base layer. Finally, we investigate the age of the CVEs that are present in our container image samples.

5.1. Vulnerabilities based on image category, attack vector and patch status

Fig. 1 highlights the number of vulnerabilities by image category across each of the three analysis periods. Overall, the number of vulnerabilities (excluding "NOFIX", but including CVEs and other security advisories such as GHSA and RHSA) increased per analysis period from 66,753 in analysis period 1 through to 84,213 in analysis period 3. This equates to an average increase of 46 vulnerabilities per image between analysis period 1 and analysis period 3. This increase was represented across all image types (Official, Verified and OSS). The lowest increase was noted in the OSS images between analysis period 2 and analysis period 3, from 16252 vulnerabilities up to 16906. Of the three image categories, it can be seen that OSS consistently had the lowest number of vulnerabilities. In analysis period 3, the total number of vulnerabilities in OSS images were 52% and 49% lower than Verified and Official images respectively.

Fig. 2 shows the number of CVEs in relation to the attack vector, which could be local, remote, or unknown, as well as whether a CVE has been labelled as "NOFIX" by the package maintainer. The prevalence of "NOFIX" CVEs ranged from 10% (analysis period 1) to 6% (analysis period 3). This likely indicates that older libraries and software which contains vulnerabilities with no fixed version are being "aged out". There were significant numbers of local CVEs which accounted for 26% (analysis period 1) to 22% (analysis period 3) of all CVEs. In total 33% to 26% of CVEs were either a "NOFIX" or a local attack vector, presenting limited or no real-world risk to the container deployment. This is especially true when a container executes as root by default, as was observed to be the case for over 70% of the images analysed, which arguably introduces further security implications beyond the stated CVEs.

Analysis of the CVEs shows that around 10% of the total number of reported CVEs are unique, highlighting the significant duplication of CVEs across all images. Table 1 shows the number of unique CVEs per analysis period. Of these unique CVEs 21% to 22% (across all 3 analysis periods and attack vectors) were either "NOFIX" or negligible severity CVEs. When looking at network attack vector only, this was 20% to 21% of reported CVEs. Combining this reduction along with the number of local attack vector CVEs would further reduce the number of potentially impactful (unique) CVEs by between 36% to 38%.

Fig. 3 investigates the nature of network, local, and unknown CVEs further by also considering the age of the CVEs (when the CVE was first reported). As can be seen, the CVE age is spread over a wide time period, with the earliest CVE dating back to 1999 (CVE-1999-0082[5]). Some minor variation is observable in the time-series plots that indicate that further CVEs may be introduced into a container image from previous years, rather than new CVEs always being from the current year. Furthermore, there is a significant increase in the number of unknown CVEs for the current year, which may be indicative of these being relatively new and so the full nature of the vulnerability has not been uncovered in full. Still, what is particularly interesting to observe is the

[5] https://www.opencve.io/cve/CVE-1999-0082.

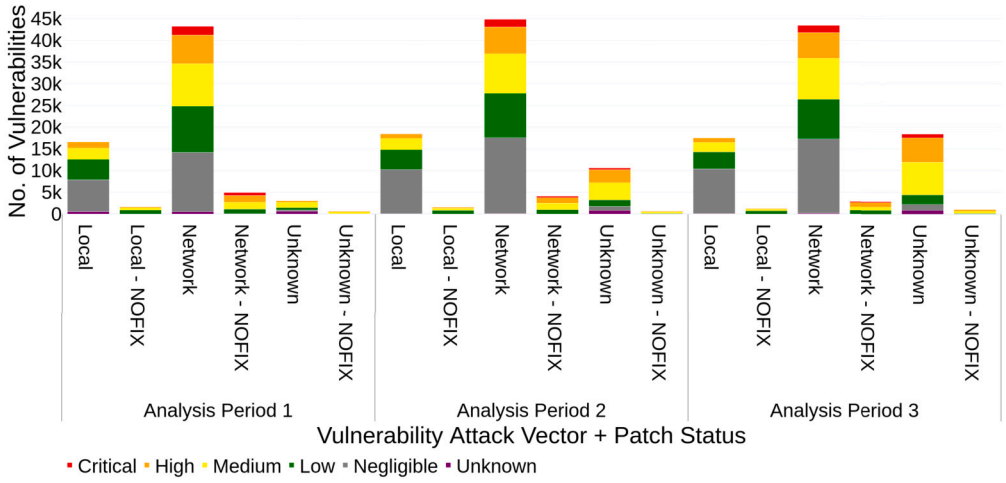


Fig. 2. Results to show the number of vulnerabilities by attack vector (local, network, unknown) and patch status (NOFIX) for each analysis period.

Table 1
Unique CVEs per analysis period (by attack vector).

Analysis Period	Attack Vector	No. of Unique CVEs	Total (Unique)	Total (Reported)
1	Network	6395		
	Local	1469	8142	74677
	Unknown	278		
2	Network	6595		
	Local	1505	9063	86368
	Unknown	963		
3	Network	6580		
	Local	1566	9526	91100
	Unknown	1380		

timeline of CVEs that remain present for years in many of the container images, despite most containers receiving regular updates and new software features.

5.2. CVEs based on image OS inheritance and container size

Fig. 4 shows the average number of vulnerabilities based on the image OS: Alpine, Debian, Ubuntu, or Linux – Other. It is clear from the bar chart that Debian images consistently exhibit a larger number of vulnerabilities compared to the other operating systems, although a large proportion of these are marked as negligible. We choose to examine this case further by considering the linkage between the image size and the number of unique CVEs.

Fig. 5 shows container size plotted against the number of unique CVEs for that container, for analysis period 1 (Fig. 5a), analysis period 2 (Fig. 5b), and analysis period 3 (Fig. 5c). Since the majority of instances cluster where CVE < 200 and size <= 1 GB we show this as a subset of the data for analysis periods 1–3 (Fig. 5d–Fig. 5f respectively). We use ordinary least squares (OLS) to show a regression trend line for each OS. We observe that Debian images have a much steeper gradient compared to other base OSs, and consistently show more CVEs compared against other container images, often including those which are similar or larger in size.

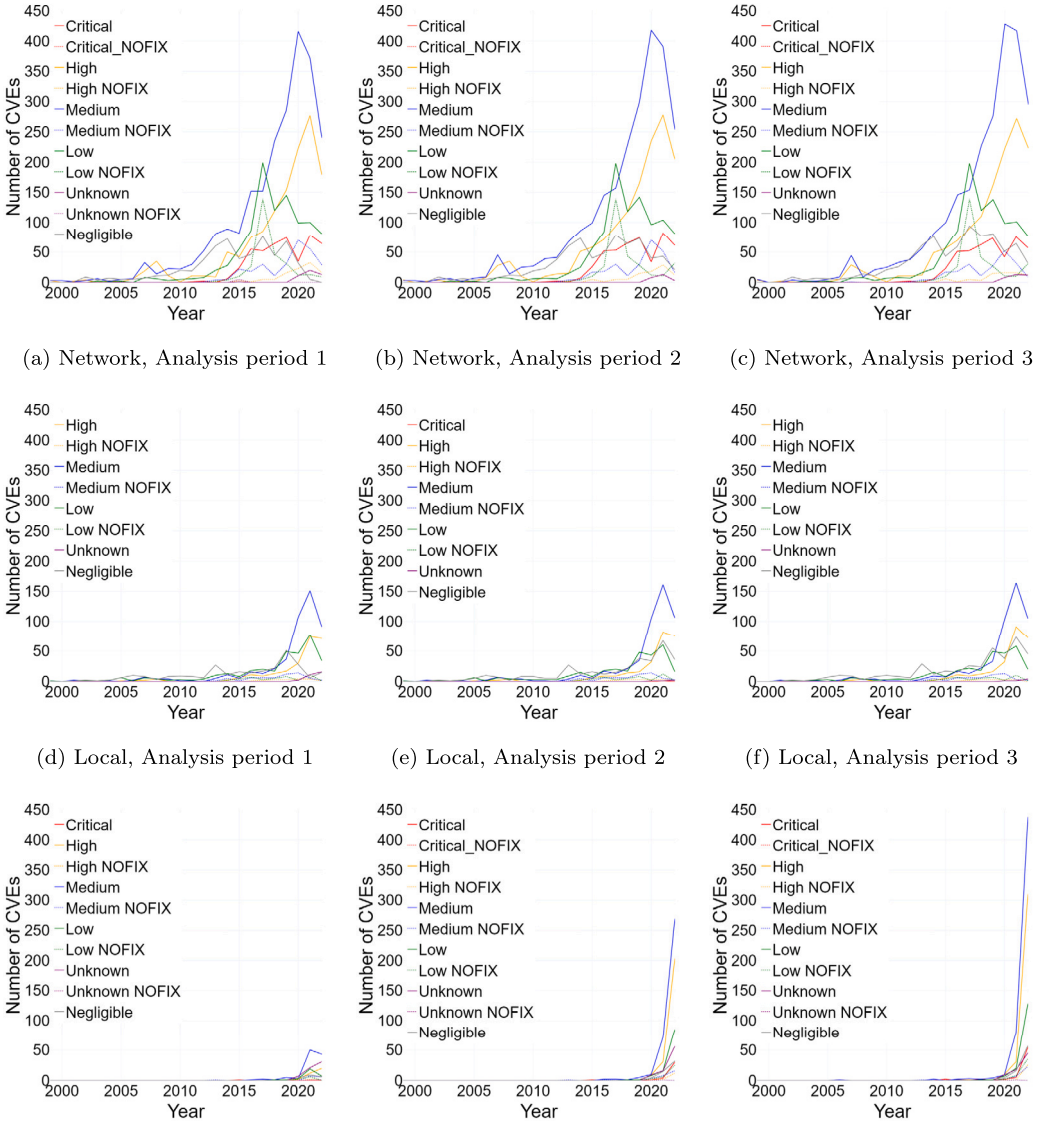
To understand the role that CVE inheritance plays within these results, further analysis was carried out looking specifically at images with shared base layers. Due to the way container images are constructed, new images can be created by "building on top" of an existing image, leading to vulnerability inheritance. We analysed base layers shared by 2 or more images only, breaking the results down in analysis periods and image OS (Alpine;Debian;Ubuntu and Linux – Other). For each shared base layer we looked at the number of images which share that base layer and then the number of total (unique) CVEs for all images with the same base layer. We then filtered out CVEs which were found in all of these images. This process was repeated for each base layer with 2 or more images. Our results can be seen in Table 2.

On average a significant portion of CVEs (70%–90%) found in images with shared base layers were not inherited, with multiple instances of 0 inherited CVEs in all 3 analysis periods. We found that comparative CVE inheritance is higher in Debian and Linux – Other images, with the highest examples of inherited CVEs being 73% and 72%, Linux – Other and Debian OS shared base layers respectively. Interestingly we found that Ubuntu images had the lowest rates of CVE inheritance. Whilst the number of shared CVEs for Ubuntu images was still high (in some cases exceeding the number of total unique CVEs in other image categories), it was comparatively small in relation to the total number of unique CVEs.

There are potentially multiple factors which contribute to this finding, including the number of available packages and the number of known CVEs for certain image OS. For example the CVE Details page for 2022 (Top 50 products having highest number of cve security vulnerabilities, 2023) list over 8000 "distinct" CVEs for Debian, less than 4000 for Ubuntu and 1377 for Opensuse (a Linux – Other OS). A full investigation into the factors behind CVE inheritance was considered outside the scope of this work, but would be of benefit to future research within this area.

5.3. Available exploits for container CVEs

Fig. 6 shows the number of CVEs by attack vector, that have existing exploits available from Exploit-DB. When looking at CVEs with exploits, we found that the highest number of images identified with at least one possible exploit was 220, which occurred in analysis period 2. This included local, network, and unknown attack vectors, at all levels of severity. When looking at images which only had remote exploits, rated higher than negligible, this dropped to a peak of 138 images in analysis period 3. The number of CVEs with exploits overall ranged from 2329 in analysis period 1 to 2613 in analysis period 3. This accounted for less the 3% of the total number of vulnerabilities per analysis period. When we look at remote exploits only (network attack vector) this became 931 in analysis period 1 to 1039 in analysis period 3, accounting for 1% of all vulnerabilities across each analysis period. This was further reduced to less than 1% when we remove exploits rated as negligible, with the highest number being 390, recorded during analysis period 2. As all exploits were taken from Exploit-DB they could be classified as either "Verified" or not. Approximately 69% of all exploits (across all 3 analysis periods) were "Verified". However it is important to note that every exploit is moderated prior to publication on Exploit-DB, and that an unverified exploit "simply means we did not have the opportunity to



(a) Network, Analysis period 1 (b) Network, Analysis period 2 (c) Network, Analysis period 3

(d) Local, Analysis period 1 (e) Local, Analysis period 2 (f) Local, Analysis period 3

(g) Unknown, Analysis period 1 (h) Unknown, Analysis period 2 (i) Unknown, Analysis period 3

Fig. 3. Unique (a)-(c) network CVEs, (d)-(f) local CVEs, and (g)-(i) unknown CVEs, by severity and age per each analysis period (values aggregated by year).

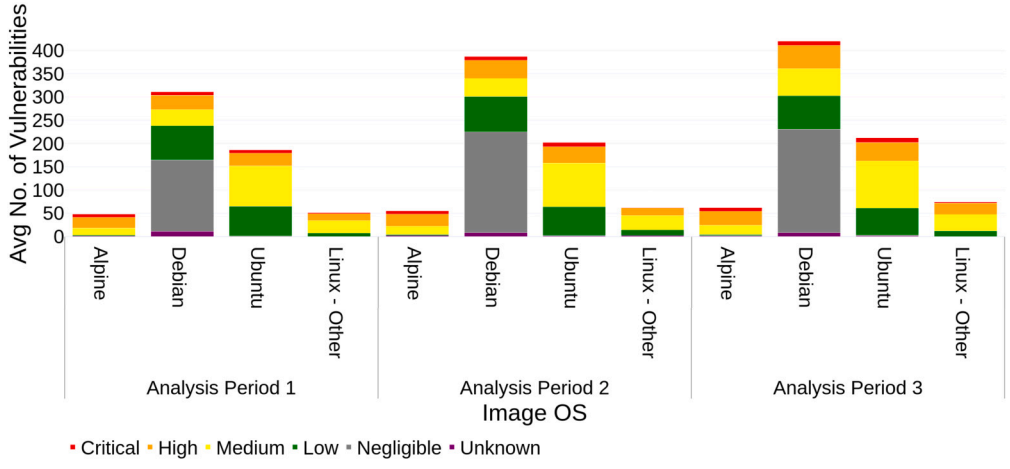


Fig. 4. Results to show the average number of vulnerabilities by image OS (Alpine, Debian, Ubuntu, Linux – Other) for each analysis period (excluding NOFIX entries).

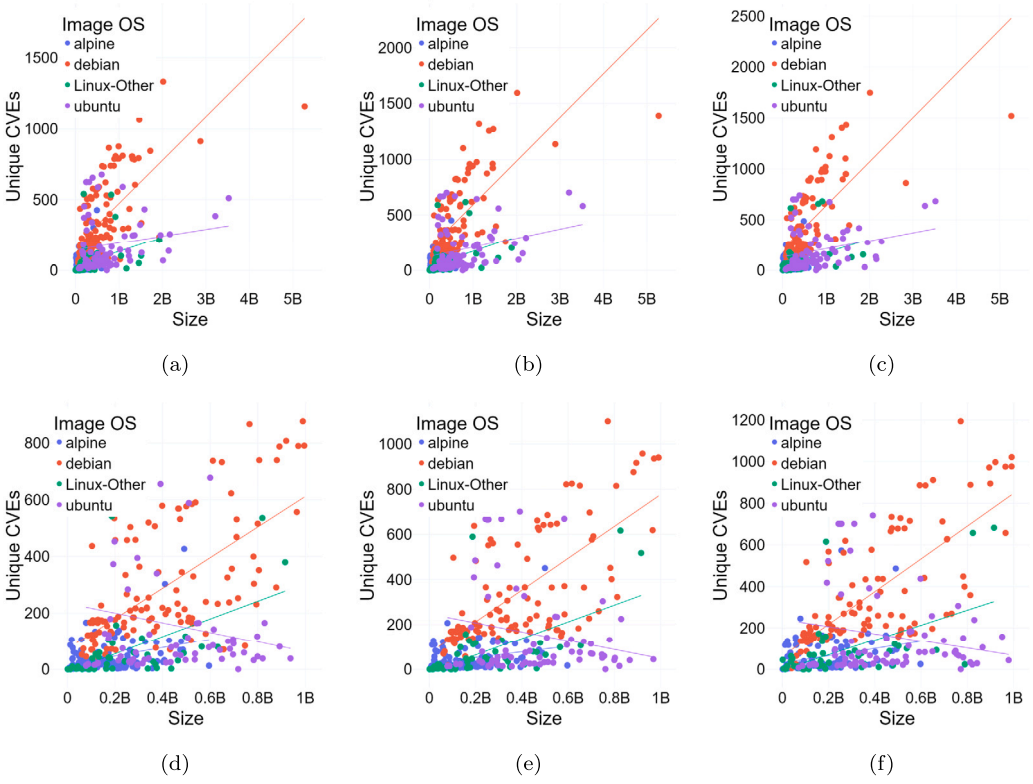


Fig. 5. Image CVEs by size and OS for (a) analysis period 1, (b) analysis period 2, and (c) analysis period 3. Plot region where CVE < 200 and size <= 100 for (d) analysis period 1, (e) analysis period 2, and (f) analysis period 3.

Table 2
Unique and shared CVEs, per image (with a shared base layer) by image OS.

Analysis Period	Image OS	No. of Shared Base Layers	Avg No. of Images (with Shared Base Layers)	Total Unique CVEs*			Shared Unique CVEs*		
				Min	Max	Mean	Min	Max	Mean
1	Alpine	13	7	9	190	97	0	32	8
	Debian	16	8	80	1531	810	0	298	66
	Ubuntu	9	7	101	844	433	0	160	22
	Linux-other	7	2	2	232	66	0	24	11
2	Alpine	14	6	19	485	125	0	51	17
	Debian	13	9	152	1818	1013	28	691	181
	Ubuntu	9	8	240	1152	532	1	394	79
	Linux-other	8	2	2	128	70	0	55	22
3	Alpine	14	6	42	498	155	0	84	14
	Debian	14	8	154	1996	1033	4	740	224
	Ubuntu	10	8	214	1179	544	2	228	49
	Linux-other	6	2	9	164	83	3	55	24

* Based on all images with a shared base layer.

test the exploit internally." (Exploit-DB, 2023). We address our decision to use Exploit-DB as our single source of truth in Section 7.

5.4. Containers with no reported vulnerabilities and additional observations

In the initial analysis period, 9 images were found to have zero vulnerabilities. This decreased to 8 in analysis period two, and only 6 in analysis period three. Of the final 6 images, it was observed that 5 of these consistently had zero vulnerabilities across all three analysis periods. These five images are shown in Table 3.

Official images were the most consistently updated, with 95% of the images being updated at least once between July 2022 and January 2023, and only 2 images not being updated at all. However, despite this the number of vulnerabilities within official images still increased per analysis period. One example case is the official *adminer*[6] image. In the initial analysis period this image had only 2 vulnerabilities (1 critical and 1 unknown), in the second analysis period this had increased to 9 and in the final analysis period it had reached 160 (1 critical, 5 high, 8 medium, 34 low, 111 negligible and 1 unknown), including 12 (all negligible) with exploits in Exploit-DB. A similar pattern was observed with other official images, including *frendica*,[7] which went from 696 vulnerabilities to 803 and finally 865, which included two high level exploits (1 local and 1 network). In the case of both *adminer* and *frendica*, updates for these images were released between the analysis stages in our study, highlighting the issue that developers are not necessarily

[6] https://hub.docker.com/_/adminer/.
[7] https://hub.docker.com/_/friendica/.

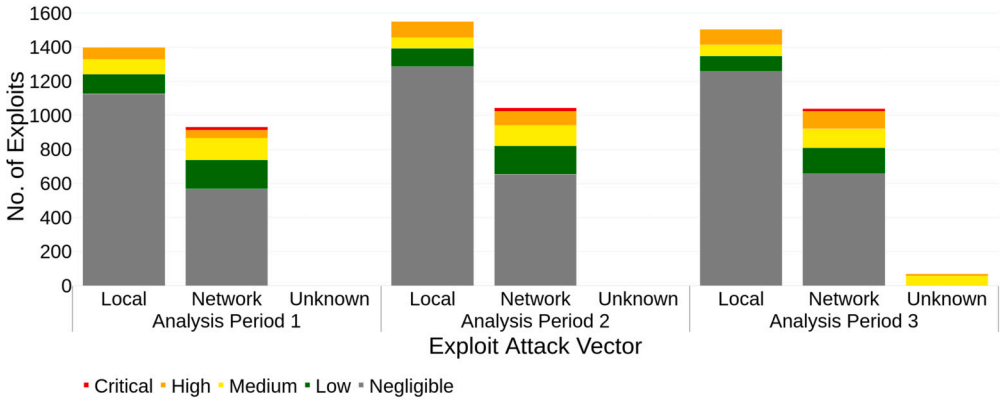


Fig. 6. Results to show the number of CVEs by attack vector (local, network, unknown) that have existing exploits available from Exploit-DB, for each analysis period.

Table 3
5 images out of 380 that exhibit zero vulnerabilities across all three analysis stages.

Image Name	Image OS	Categorisation
alt	Altlinux	Offical
buildpack-deps	Ubuntu	Offical
clearlinux/iperf	Clear Linux	OSS
clearlinux/redis	Clear Linux	OSS
docker/ecs-searchdomain-sidecar	Linux-Unknown	Verified

screening containers for the possibility of vulnerabilities, and in some cases, applied updates or additional software functionality are actively decreasing the security posture of container images.

6. Discussion

Taking an overview of the complete study, it can be observed that container image security is continually changing across the 3 distinct observation periods, highlighting the fact that a single time point, as often used in previous works, can not adequately represent an understanding of the evolving security landscape over time. From the three analysis periods, whilst some cases of unknown, negligible or low vulnerabilities did decrease, overall the number of vulnerabilities reported in our set of 380 container images increased from 66,753 in July 2022 to 84,213 in January 2023, an increase of 26.1%.

Whilst this in itself is a concerning finding, it should also be noted that the large number of vulnerabilities reported are not necessarily reflective of what may be considered the real-world risk or impact. Only 10% of the reported CVEs were deemed unique indicating significant duplication, and around 36% to 38% of the unique CVEs could be considered as low real-world impact or risk due to a combination of patch status ("NOFIX"), severity (negligible) and attack vector (local). In total, the number of all reported vulnerabilities could be reduced by as much as 33% when omitting those that are based on a local attack vector or that have been deemed as "NOFIX" by the maintainers. An alternative analysis would be on vulnerabilities which have known exploits, reducing the number of vulnerabilities that could be considered as high risk, with a potential real-world impact, down to as little as 1% or less of the initially reported numbers. Our study has begun to systematically highlight these issues, both in terms of the number of reported vulnerabilities, and the associated factors that may influence whether the vulnerability is significant enough to warrant concern.

By carrying out multiple different avenues of vulnerability analyses during our study, which included factors such as image size, image category and image OS we have been able to highlight which contributions have the most significant impact on container image security posture. For example our analysis has shown that Debian images consistently have higher numbers of reported vulnerabilities both overall and when focusing on other factors such as image size and image inheritance. However this may be due to the much higher number of vulnerabilities reported against this OS when compared to other image OS (Top 50 products having highest number of cve security vulnerabilities, 2023). Whether this is due to more active security research and reporting within the Debian community, a larger number of packages (and therefore wider attack surface) by default or that other OS do in fact have fewer vulnerabilities would need to be investigated further.

For the container images in our sample, we observed a significant spread in CVE age, with some CVEs dating as far back as 1999 still present in some images today, including critical CVEs with fixes. It is possible that these have not been removed or addressed as they have been assessed as no threat or impact by the maintainers. An example case would be CVE-2005-2541.[8] This is a critical, remote CVE that impacts tar (1.15.1), specifically during file extract. However, if the container or associated micro-service does not allow for user file extraction then the vulnerability is actually mitigated by the container use case. In either case our study has highlighted that active development and maintenance of images does not itself ensure an improved security posture, with an overall increase in container vulnerabilities found during the course of our longitudinal study and multiple instances of images having additional vulnerabilities and exploits despite being updated.

There is clearly a growing need to ensure that publicly-available containers are provided securely, without posing a vulnerability to their particular usage. However, it is also apparent that the risk that containers present is not accurately reflected by many of the previous reports and analysis that have been conducted, where the focus has been on the number, severity and age of vulnerabilities. By considering the potential real-world impact across a longitudinal series of analysis periods we showcase that a significant portion of the vulnerabilities reported present little to no risk to a running container environment, especially in consideration of access (local vulnerabilities), running environment (containers which run as a root user by default) and existence of a working exploit. What is perhaps of far more concern and impact to the security posture of a containerised environment is the initial setup and configuration. Mounted volumes, over-privileged containers, default root users and exposed orchestration ports, all provide potential attack vectors for container escape with root privilege onto the host system.

7. Conclusions and further work

Our study has shown that the number of vulnerabilities, present in all categories, increased during the period of July 2022 to January 2023. This increase was not limited or directly tied into a lack of im-

[8] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-2541.

age updates, with instances of images being updated per analysis run showing an increase in both vulnerabilities and exploits. However, a very small number of these, less than 1% of all vulnerabilities, per analysis period, present what could be considered as a high risk, potentially real-world impact, based on a combination of attack vector, severity and the existence of an exploit (within Exploit-DB (2023)).

Whilst previous studies have attempted to analyse a greater number of containers than in our study, our focus is on acquiring a deeper understanding of the longitudinal security analysis of the container landscape. We have selected containers to ensure that our sample is representative of the common and popular images used within DockerHub, so that the output of this study provides significant detail and context-based analysis that is currently not presented in existing surveys.

Our exploit metric is currently based on a single source of truth (Exploit-DB, 2023) and as such it is possible that some vulnerabilities have publicly available exploits on platforms such as GitHub. However, as not all exploits listed on a platform like GitHub can necessarily be trusted (AutoHoneyPoC, 2023), it would not be feasible to assess and verify whether such exploits are "genuine". The use of moderation before exploit publication provided by Exploit-DB provides a degree of oversight that is missing from sources such as Github. Understanding the extent of open-source exploits was considered as out of scope for this paper, however we leave as a open challenge for the reader and the research community.

Future work could extend our analysis to assess whether the container and micro-service use cases actually mitigate any existing vulnerabilities that may be present, such that they are not able to be exploited. This would require that the micro-services are run through end-to-end, including edge case and error based scenarios, and that all syscalls monitored. While previous works have looked at syscall monitoring within micro-services (Ghavamnia et al., 2020) this has been with a focus on kernel vulnerabilities and preventing runtime escapes. This could potentially be adapted to assist with the real-world impact analysis of container vulnerabilities. As the nature of containers and their usage continues to increase, this deeper investigation into container security would help to understand how security continues to be managed across publicly-available container deployment.

CRediT authorship contribution statement

Alan Mills: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing. **Jonathan White:** Conceptualization, Investigation, Methodology, Validation, Writing – review & editing. **Phil Legg:** Conceptualization, Methodology, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data has been made available via a public github repo. Link included in the submitted manuscript.

Acknowledgements

This work was supported by the College of Arts, Technology and Environment at the University of the West of England.

Appendix A. Image tags across analysis periods

For the below tables the following key is used:

- Different image (hash) and tag
- Different image (hash) with the same tag
- The same image (hash) and tag

A.1. Official images

Analysis Period 1	Analysis Period 2	Analysis Period 3
adminer-4.8.1	adminer-4.8.1	adminer-4.8.1
aerospike-ce-6.0.0.1	aerospike-ce-6.1.0.2	aerospike-ce-6.2.0.2
almalinux-9.0	almalinux-9.0	almalinux-9.1
alpine-3.16.0	alpine-3.16	alpine-3.17.0
alt-p9	alt-p9	alt-p10
amazoncorretto-18.0.1	amazoncorretto-18.0.2	amazoncorretto-19.0.1
amazonlinux-2022	amazonlinux-2022	amazonlinux-2022
api-firewall-0.6.8	api-firewall-0.6.9	api-firewall-0.6.9
archlinux-base	archlinux-base	archlinux-base
backdrop-1.21.4	backdrop-1.23.0	backdrop-1.23.1
bash-5.1	bash-5.2.0	bash-5.2.15
bonita-7.13.0	bonita-7.13.0	bonita-7.13.0
buildpack-deps-22.10	buildpack-deps-22.10	buildpack-deps-23.04
busybox-1.35.0	busybox-1.35	busybox-1.35.0
caddy-2.5.1	caddy-2.6.1	caddy-2.6.2
cassandra-4.0.4	cassandra-4.0.6	cassandra-4.1.0
chronograf-1.9.4	chronograf-1.9.4	chronograf-1.9.4
cirros-0.5.2	cirros-0.6.0	cirros-0.6.1
clojure-temurin-18-lein-2.9.8-alpine	clojure-temurin-19-lein-2.9.10-alpine	clojure-temurin-8-lein-2.10.0-alpine
composer-2.3.7	composer-2.4.2	composer-2.5.1
consul-1.12.2	consul-1.12.5	consul-1.14.3
convertigo-8.0.0	convertigo-8.0.2	convertigo-8.1.0
couchbase-community-7.1.0	couchbase-community-7.1.1	couchbase-community-7.1.1
couchdb-3.2.2	couchdb-3.2.2	couchdb-3.2.2
crate-4.8.1	crate-4.8.4	crate-5.1.2
dart-2.17.3	dart-2.18.2	dart-2.18.6
debian-stable	debian-stable	debian-stable
docker-20.10.17	docker-20.10.18	docker-20.10.22
drupal-9.2.20-php7.4	drupal-9.4.7-php7.4	drupal-9.4.8-php7.4
eclipse-mosquitto-2.0.14	eclipse-mosquitto-2.0.15	eclipse-mosquitto-2.0.15
eclipse-temurin-8	eclipse-temurin-8	eclipse-temurin-8
eggdrop-1.9.2	eggdrop-1.9.3	eggdrop-1.9.4
elasticsearch-8.2.2	elasticsearch-8.4.2	elasticsearch-8.5.3
elixir-1.13.4	elixir-1.13.4	elixir-1.13.4
erlang-25	erlang-25	erlang-25.2.0
fedora-36	fedora-36	fedora-38
flink-1.15.0	flink-1.15.2	flink-1.15.3
fluentd-v1.14.0-1.0	fluentd-v1.14.0-1.0	fluentd-v1.14.0-1.0
friendica-stable	friendica-stable	-stable
gazebo-libgazebo11	gazebo-libgazebo11	gazebo-libgazebo11
gcc-9.5.0	gcc-9.5.0	gcc-9.5.0
geonetwork-4.2.0	geonetwork-4.2.1	geonetwork-4.2.2
ghost-5.2.2	ghost-5.16.2	ghost-5.26.3
golang-1.18.3	golang-1.18.6	golang-1.18.9
gradle-7.4.2	gradle-7.5.1	gradle-7.6
groovy-4.0.3	groovy-4.0.5	groovy-4.0.7
haproxy-2.6.0	haproxy-2.6.6	haproxy-2.7.1
hitch-1.7.2-1	hitch-1.7.3-1	hitch-1.7.3-1
httpd-2.4.54	httpd-2.4.54	httpd-2.4.54
hylang-python3.10	hylang-python3.10	hylang-python3.10
ibmjava-8	ibmjava-8	ibmjava-8
ibm-semeru-runtimes-open-8-jdk	ibm-semeru-runtimes-open-8-jdk	ibm-semeru-runtimes-open-8-jdk
ibm-semeru-runtimes-open-8-jre	ibm-semeru-runtimes-open-8-jre	ibm-semeru-runtimes-open-8-jre
influxdb-2.2.0	influxdb-2.4.0	influxdb-2.6.0
irssi-1.2.3	irssi-1.4.2	irssi-1.4.3
jetty-11.0.9	jetty-11.0.12	jetty-11.0.13
joomla-3.10.9	8joomla-3.10.11	joomla-3.10.11
jruby-9.3.4.0	jruby-9.3.8.0	jruby-9.4.0.0
julia-1.7.3	julia-1.8.1	julia-1.8.4
kapacitor-1.5.9	kapacitor-1.5.9	kapacitor-1.5.9
kibana-7.17.4	kibana-7.17.6	kibana-7.17.8
kong-2.8.1	kong-3.0.0	kong-3.1.1
lightstreamer-7.2.2	lightstreamer-7.2.2	lightstreamer-7.3.2
logstash-7.17.4	logstash-7.17.6	logstash-7.17.8
mageia-8	mageia-8	mageia-8
mariadb-10.7.4	mariadb-10.7.6	mariadb-10.9.4
matomo-4.10.1	matomo-4.11.0	matomo-4.13.0
maven-3.8.5	maven-3.8.6	maven-3.8.6
mediawiki-1.38.1	mediawiki-1.38.2	mediawiki-1.38.5
memcached-1.6.15	memcached-1.6.17	memcached-1.6.17
mongo-5.0.9	mongo-5.0.13	mongo-5.0.14
mongo-express-0.54.0	mongo-express-0.54.0	mongo-express-0.54.0
monica-3.7.0	monica-3.7.0	monica-3.7.0
mono-6.12.0.122	mono-6.12.0.182	mono-6.12.0.182
mysql-8.0.29	mysql-8.0.30	mysql-8.0.31

nats-2.8.4	nats-2.9.2	nats-2.9.10
neo4j-4.4.7	neo4j-4.4.9	neo4j-4.4.16
neurodebian-nd20.04	neurodebian-nd22.04	neurodebian-nd22.04
nextcloud-24.0.1	nextcloud-24.0.5	nextcloud-24.0.8
nginx-1.22.0	nginx-1.22.0	nginx-1.22.1
node-18.3.0	node-18.10.0	node-18.12.1
notary-server-0.7.0	notary-server-0.7.0	notary-server-0.7.0
notary-signer-0.7.0	notary-signer-0.7.0	notary-signer-0.7.0
odoo-15.0	odoo-15.0	odoo-15.0
openjdk-19-jdk	openjdk-19-jdk	openjdk-21-jdk
open-liberty-full	open-liberty-full	open-liberty-full
orientdb-3.2.6	orientdb-3.2.10	orientdb-3.2.13
percona-8.0.28-19	percona-8.0.29-21	percona-8.0.29-21
perl-5.36.0	perl-5.36.0	perl-5.36.0
photon-4.0-20220603	photon-4.0-20220923	photon-4.0-20221224
php-7.4.30	php-7.4.32	php-7.4.33
phpmyadmin-5.2.0	phpmyadmin-5.2.0	phpmyadmin-5.2.0
php-zendserver-2021.0	php-zendserver-2021.0	php-zendserver-2021.0
plone-5.2.7	plone-5.2.9	plone-5.2.9
postfixadmin-3.3.11	postfixadmin-3.3.11	postfixadmin-3.3.13
postgres-14.3	postgres-14.5	postgres-14.6
pypy-3.9-7.3.9	pypy-3.9-7.3.9	pypy-3.9-7.3.10
python-3.10.5	python-3.10.7	python-3.10.9
rabbitmq-3.9.20	rabbitmq-3.9.22	rabbitmq-3.9.27
rakudo-star-2021.04	rakudo-star-2022.07	rakudo-star-2022.12
redis-7.0.1	redis-7.0.5	redis-7.0.7
registry-2.8.1	registry-2.8.1	registry-2.8.1
rethinkdb-2.4.2	rethinkdb-2.4.2	rethinkdb-2.4.2
rocket.chat-4.6.3	rocket.chat-4.8.4	rocket.chat-4.8.7
rockylinux-8	rockylinux-9.0	rockylinux-9.1
ros-noetic	ros-noetic	ros-noetic
rust-1.61.0	rust-1.64.0	rust-1.66.0
sapmachine-18.0.1.1	sapmachine-18.0.2.1	sapmachine-19.0.1
silverpeas-6.2.3	silverpeas-6.2.3	silverpeas-6.3
sl-7	sl-7	sl-7
solr-9.0.0	solr-9.0	solr-9.1.0
sonarqube-lts	sonarqube-lts	sonarqube-lts
spiped-1.6.2	spiped-1.6.2	spiped-1.6.2
storm-2.4.0	storm-2.4.0	storm-2.4.0-temurin
swift-5.6.1	swift-5.6.3	swift-5.7.2
swipl-8.5.10	swipl-8.5.17	swipl-8.5.20
teamspeak-3.13.6	teamspeak-3.13.7	teamspeak-3.13.7
telegraf-1.22.4	telegraf-1.22.4	telegraf-1.22.4
tomcat-9.0.64	tomcat-9.0.67	tomcat-9.0.70
traefik-v2.7.0	traefik-v2.8	traefik-v2.9.6
ubuntu-22.04	ubuntu-22.04	ubuntu-22.10
varnish-7.0.2	varnish-7.0.3	varnish-7.1.2
vault-1.9.6	vault-1.9.9	vault-1.10.9
websphere-liberty-full	websphere-liberty-full	websphere-liberty-full
wordpress-php8.0	wordpress-php8.1	wordpress-php8.1
xwiki-14.4.1	xwiki-14.4.4	xwiki-14.4.7
yourls-1.9	yourls-1.9.1	yourls-1.9.1
znc-1.8.2	znc-1.8.2	znc-1.8.2
zookeeper-3.7.1	zookeeper-3.7.1	zookeeper-3.7.1-temurin

A.2. Verified images

Analysis Period 1	Analysis Period 2	Analysis Period 3
amazon/amazon-ecs-agent-v1.61.3	amazon/amazon-ecs-agent-v1.63.1	amazon/amazon-ecs-agent-v1.67.2
amazon/aws-alb-ingress-controller-v2.4.2	amazon/aws-alb-ingress-controller-v2.4.4	amazon/aws-alb-ingress-controller-v2.4.5
amazon/aws-cli-2.7.12	amazon/aws-cli-2.8.0	amazon/aws-cli-2.9.13
amazon/aws-efs-csi-driver-v1.4.0	amazon/aws-efs-csi-driver-v1.4.2	amazon/aws-efs-csi-driver-v1.4.9
amazon/aws-for-fluent-bit-2.26.0	amazon/aws-for-fluent-bit-2.28.1	amazon/aws-for-fluent-bit-2.29.0
amazon/aws-node-termination-handler-v1.12.1	amazon/aws-node-termination-handler-v1.12.1	amazon/aws-node-termination-handler-v1.12.1
amazon/aws-xray-daemon-3.3.3	amazon/aws-xray-daemon-3.3.5	amazon/aws-xray-daemon-3.3.5
amazon/cloudwatch-agent-1.247352.0b251908	amazon/cloudwatch-agent-1.247355.0b252062	amazon/cloudwatch-agent-1.247357.0b252275
amazon/dynamodb-local-1.18.0	amazon/dynamodb-local-1.20.0	amazon/dynamodb-local-1.20.0
atlassian/bitbucket-server-8.1.0	atlassian/bitbucket-server-8.1.0	atlassian/bitbucket-server-8.1.0
atlassian/confluence-server-7.16	atlassian/confluence-server-7.19.1	atlassian/confluence-server-7.19.4
atlassian/default-image-3	atlassian/default-image-3	atlassian/default-image-3
atlassian/pipelines-auth-proxy-prod	atlassian/pipelines-auth-proxy-prod	atlassian/pipelines-auth-proxy-prod
atlassian/pipelines-docker-daemon-v20-stable	atlassian/pipelines-docker-daemon-v20-stable	atlassian/pipelines-docker-daemon-v20-stable
atlassian/pipelines-dvcstools-prod-stable	atlassian/pipelines-dvcstools-prod-stable	atlassian/pipelines-dvcstools-prod-stable
balena/armv7hf-supervisor-v12.11.35	balena/armv7hf-supervisor-v12.11.35	balena/armv7hf-supervisor-v12.11.35
bitnami/alertmanager-0.24.0	bitnami/alertmanager-0.24.0	bitnami/alertmanager-0.24.0
bitnami/cassandra-4.0.4	bitnami/cassandra-4.0.6	bitnami/cassandra-4.0.7
bitnami/dokuwiki-20200729	bitnami/dokuwiki-20220731	bitnami/dokuwiki-20220731

bitnami/elasticsearch-8.2.3	bitnami/elasticsearch-8.4.2	bitnami/elasticsearch-8.5.3
bitnami/etcd-3.5.4	bitnami/etcd-3.5.5	bitnami/etcd-3.5.6
bitnami/external-dns-0.12.0	bitnami/external-dns-0.12.2	bitnami/external-dns-0.13.1
bitnami/fluentd-1.15.0	bitnami/fluentd-1.15.2	bitnami/fluentd-1.15.3
bitnami/ghost-5.2.3	bitnami/ghost-5.17.0	bitnami/ghost-5.28.0
bitnami/grafana-8.5.6	bitnami/grafana-8.5.10	bitnami/grafana-9.3.2
bitnami/kafka-3.2.0	bitnami/kafka-3.2.3	bitnami/kafka-3.2.3
bitnami/kube-state-metrics-2.5.0	bitnami/kube-state-metrics-2.6.0	bitnami/kube-state-metrics-2.7.0
bitnami/mariadb-10.8.3	bitnami/mariadb-10.8.5	bitnami/mariadb-10.8.6
bitnami/mariadb-galera-10.8.3	bitnami/mariadb-galera-10.8.4	bitnami/mariadb-galera-10.8.6
bitnami/memcached-1.6.15	bitnami/memcached-1.6.17	bitnami/memcached-1.6.17
bitnami/metrics-server-0.6.1	bitnami/metrics-server-0.6.1	bitnami/metrics-server-0.6.2
bitnami/minideb-bullseye	bitnami/minideb-bullseye	bitnami/minideb-bullseye
bitnami/mongodb-5.0.9	bitnami/mongodb-5.0.13	bitnami/mongodb-5.0.14
bitnami/mongodb-exporter-0.32.0	bitnami/mongodb-exporter-0.34.0	bitnami/mongodb-exporter-0.36.0
bitnami/mysql-8.0.29	bitnami/mysql-8.0.30	bitnami/mysql-8.0.31
bitnami/nginx-1.23.0	bitnami/nginx-1.23.1	bitnami/nginx-1.23.3
bitnami/node-18.4.0	bitnami/node-18.10.0	bitnami/node-18.13.0
bitnami/node-exporter-1.3.1	bitnami/node-exporter-1.3.1	bitnami/node-exporter-1.5.0
bitnami/oauth2-proxy-7.3.0	bitnami/oauth2-proxy-7.3.0	bitnami/oauth2-proxy-7.4.0
bitnami/phpmyadmin-5.2.0	bitnami/phpmyadmin-5.2.0	bitnami/phpmyadmin-5.2.0
bitnami/postgres-exporter-0.10.1	bitnami/postgres-exporter-0.11.1	bitnami/postgres-exporter-0.11.1
bitnami/postgresql-14.4.0	bitnami/postgresql-14.5.0	bitnami/postgresql-14.6.0
bitnami/postgresql-repmgr-13.7.0	bitnami/postgresql-repmgr-13.8.0	bitnami/postgresql-repmgr-13.9.0
bitnami/prometheus-2.36.2	bitnami/prometheus-2.38.0	bitnami/prometheus-2.41.0
bitnami/rabbitmq-3.10.5	bitnami/rabbitmq-3.10.8	bitnami/rabbitmq-3.10.13
bitnami/redis-7.0.2	bitnami/redis-7.0.5	bitnami/redis-7.0.5
bitnami/redis-exporter-1.43.0	bitnami/redis-exporter-1.44.0	bitnami/redis-exporter-1.45.0
bitnami/redis-sentinel-7.0.2	bitnami/redis-sentinel-7.0.5	bitnami/redis-sentinel-7.0.5
bitnami/wordpress-6.0.0	bitnami/wordpress-6.0.2	bitnami/wordpress-6.1.1
bitnami/zookeeper-3.8.0	bitnami/zookeeper-3.8.0	bitnami/zookeeper-3.8.0
cimg/base-2022.06	cimg/base-2022.09	cimg/base-2023.01
cimg/node-18.4.0	cimg/node-18.9	cimg/node-18.12.1
cimg/python-3.10.5	cimg/python-3.10.7	cimg/python-3.10.9
cimg/ruby-3.1.2	cimg/ruby-3.1.2	cimg/ruby-3.1.3
circleci/android-api-29	circleci/android-api-29	circleci/android-api-29
circleci/docker-gc-2.0	circleci/docker-gc-2.0	circleci/docker-gc-2.0
circleci/frontend-0.2.26991-11cd543	circleci/frontend-0.2.28621-967c0c0	circleci/frontend-0.2.29418-e045dc0
circleci/golang-1.17.5	circleci/golang-1.17.5	circleci/golang-1.17.5
circleci/mongo-5.0.6	circleci/mongo-5.0.6	circleci/mongo-5.0.6
circleci/mysql-8.0.28	circleci/mysql-8.0.28	circleci/mysql-8.0.28
circleci/node-16.13.1	circleci/node-16.13.1	circleci/node-16.13.1
circleci/openjdk-17-jdk-buster	circleci/openjdk-17-jdk-buster	circleci/openjdk-17-buster
circleci/php-7.4.27	circleci/php-7.4.27	circleci/php-7.4.27
circleci/postgres-13.5	circleci/postgres-13.5	circleci/postgres-13.5
circleci/python-3.10.1	circleci/python-3.10.1	circleci/python-3.10.1
circleci/redis-7.0-rc	circleci/redis-7.0-rc	circleci/redis-7.0-rc
circleci/ruby-3.0.3	circleci/ruby-3.0.3	circleci/ruby-3.0.3
cockroachdb/cockroach-v22.1.2	cockroachdb/cockroach-v22.1.8	cockroachdb/cockroach-v22.1.12
datadog/agent-6	datadog/agent-6	datadog/agent-6
datadog/cluster-agent-1.21.0	datadog/cluster-agent-1.22.0	datadog/cluster-agent-7.41.1
datadog/docker-dd-agent-12.9.5328	datadog/docker-dd-agent-12.9.5328	datadog/docker-dd-agent-12.9.5328
docker/aci-hostnames-sidecar-1.0	docker/aci-hostnames-sidecar-1.0	docker/aci-hostnames-sidecar-1.0
docker/dockerfile-1.4.2	docker/dockerfile-1.4.3	docker/dockerfile-1.4.3
docker/ecs-searchdomain-sidecar-latest	docker/ecs-searchdomain-sidecar-latest	docker/ecs-searchdomain-sidecar-latest
docker/ucp-3.1.14	docker/ucp-3.1.14	docker/ucp-3.1.14
docker/ucp-agent-3.1.14	docker/ucp-agent-3.1.14	docker/ucp-agent-3.1.14
docker/ucp-auth-3.1.14	docker/ucp-auth-3.1.14	docker/ucp-auth-3.1.14
docker/ucp-interlock-3.1.14	docker/ucp-interlock-3.1.14	docker/ucp-interlock-3.1.14
docker/ucp-interlock-proxy-3.1.14	docker/ucp-interlock-proxy-3.1.14	docker/ucp-interlock-proxy-3.1.14
dynatrace/oneagent-1.65.1000	dynatrace/oneagent-1.68.1000	dynatrace/oneagent-1.71.1000
google/cloud-sdk-392.0.0	google/cloud-sdk-404.0.0	google/cloud-sdk-412.0.0
grafana/agent-main-2ead6ed	grafana/agent-main-a2ee071	grafana/agent-main-d62869d
grafana/fluent-bit-plugin-loki-main-25912ea-amd64	grafana/fluent-bit-plugin-loki-main-8886800-amd64	grafana/fluent-bit-plugin-loki-main-785fc2a-amd64
grafana/grafana-9.0.2	grafana/grafana-9.0.9	grafana/grafana-9.3.2
grafana/loki-main-25912ea	grafana/loki-main-8886800	grafana/loki-main-785fc2a
grafana/promtail-main-92aa69b	grafana/promtail-main-8886800	grafana/promtail-main-785fc2a
ibmcom/ibm-common-service-catalog-3.13	ibmcom/ibm-common-service-catalog-3.13	ibmcom/ibm-common-service-catalog-3.13
lacework/datacollector-5.7.0	lacework/datacollector-6.0.2	lacework/datacollector-6.2.0
mirantis/ucp-agent-3.4.10	mirantis/ucp-agent-3.4.11	mirantis/ucp-agent-3.4.12
mirantis/ucp-auth-3.4.10	mirantis/ucp-auth-3.4.11	mirantis/ucp-auth-3.4.12
newrelic/infrastructure-1.27.4	newrelic/infrastructure-1.31.0	newrelic/infrastructure-1.36.0
newrelic/infrastructure-bundle-2.8.20	newrelic/infrastructure-bundle-2.8.31	newrelic/infrastructure-bundle-2.8.37
newrelic/infrastructure-k8s-2.13.1	newrelic/infrastructure-k8s-2.13.3	newrelic/infrastructure-k8s-2.13.5
newrelic/nrsysmond-2.3.0.132	newrelic/nrsysmond-2.3.0.132	newrelic/nrsysmond-2.3.0.132
newrelic/php-daemon-10.0.0	newrelic/php-daemon-10.1.0	newrelic/php-daemon-10.4.0

percona/percona-xtradb-cluster-operator-1.11.0	percona/percona-xtradb-cluster-operator-1.11.0	percona/percona-xtradb-cluster-operator-1.13.0
portainer/agent-2.14.0	portainer/agent-2.15.1	portainer/agent-2.16.2
portainer/portainer-ce-2.14.0	portainer/portainer-ce-2.15.1	portainer/portainer-ce-2.16.2
puppet/continuous-delivery-for-puppet-enterprise-3.13.8	puppet/continuous-delivery-for-puppet-enterprise-3.13.8	puppet/continuous-delivery-for-puppet-enterprise-3.13.8
puppet/puppetserver-7.8.0	puppet/puppetserver-7.8.0	puppet/puppetserver-7.9.2
rancher/agent-v1.2.11	rancher/agent-v1.2.11	rancher/agent-v1.2.11
rancher/calico-cni-v3.18.1	rancher/calico-cni-3.21.1-rancher1	rancher/calico-cni-v3.24.1-rancher1
rancher/calico-node-v3.18.1	rancher/calico-node-v3.18.1	rancher/calico-node-v3.18.1
rancher/coredns-coredns-1.8.3	rancher/coredns-coredns-1.8.3	rancher/coredns-coredns-1.8.3
rancher/coreos-flannel-v0.13.0-rancher1	rancher/coreos-flannel-v0.13.0-rancher1	rancher/coreos-flannel-v0.13.0-rancher1
rancher/dns-v0.17.4	rancher/dns-v0.17.4	rancher/dns-v0.17.4
rancher/fleet-agent-v0.3.9	rancher/fleet-agent-v0.3.11	rancher/fleet-agent-v0.4.1
rancher/fleet-v0.3.10-rc1-linux-amd64	rancher/fleet-v0.3.11-rc1-linux-amd64	rancher/fleet-v0.6.0-rc.1-linux-amd64
rancher/gitjob-v0.1.28	rancher/gitjob-v0.1.32	rancher/gitjob-v0.1.35
rancher/healthcheck-v0.3.8	rancher/healthcheck-v0.3.8	rancher/healthcheck-v0.3.8
rancher/hyperkube-v1.20.15-rancher2	rancher/hyperkube-v1.22.15-rancher1	rancher/hyperkube-v1.22.17-rancher1
rancher/istio-proxyv2-1.8.3	rancher/istio-proxyv2-1.8.3	rancher/istio-proxyv2-1.8.3
rancher/klipper-lb-v0.3.5	rancher/klipper-lb-v0.3.5	rancher/klipper-lb-v0.4.0
rancher/lb-service-haproxy-v0.9.14	rancher/lb-service-haproxy-v0.9.14	rancher/lb-service-haproxy-v0.9.14
rancher/local-path-provisioner-v0.0.22	rancher/local-path-provisioner-v0.0.22	rancher/local-path-provisioner-v0.0.23
rancher/metadata-v0.10.4	rancher/metadata-v0.10.4	rancher/metadata-v0.10.4
rancher/metrics-server-v0.4.1	rancher/metrics-server-v0.4.1	rancher/metrics-server-v0.4.1
rancher/net-v0.13.17	rancher/net-v0.13.17	rancher/net-v0.13.17
rancher/network-manager-v0.7.22	rancher/network-manager-v0.7.22	rancher/network-manager-v0.7.22
rancher/nginx-ingress-controller-nginx-1.2.1-hardened7	rancher/nginx-ingress-controller-nginx-1.2.1-hardened7	rancher/nginx-ingress-controller-nginx-1.4.1-hardened2
rancher/nginx-ingress-controller-nginx-1.2.1-rancher1	rancher/nginx-ingress-controller-nginx-1.2.1-rancher1	rancher/nginx-ingress-controller-nginx-1.2.1-rancher1
rancher/os-3cf4bdb-amd64	rancher/os-3cf4bdb-amd64	rancher/os-3cf4bdb-amd64
rancher/rancher-agent-v2.6.6	rancher/rancher-agent-v2.6.8	rancher/rancher-agent-v2.6.9
rancher/rancher-v2.6-head	rancher/rancher-v2.6-head	rancher/rancher-v2.6-head
rancher/rke-tools-v0.1.83	rancher/rke-tools-v0.1.87	rancher/rke-tools-v0.1.88
rancher/server-v1.6.30	rancher/server-v1.6.30	rancher/server-v1.6.30
rancher/shell-v0.1.19	rancher/shell-v0.1.19-rc4	rancher/shell-v0.1.19-rc7
sysdig/agent-12.7.0	sysdig/agent-12.8.1	sysdig/agent-12.10.1
wallarm/api-firewall-v0.6.8	wallarm/api-firewall-v0.6.9	wallarm/api-firewall-v0.6.10

A.3. OSS images

Analysis Period 1	Analysis Period 2	Analysis Period 3
curlimages/curl-7.84.0	curlimages/curl-7.85.0	curlimages/curl-7.87.0
linuxserver/radarr-4.1.0	linuxserver/radarr-4.1.0	linuxserver/radarr-4.2.4
linuxserver/sonarr-3.0.8	linuxserver/sonarr-3.0.9	linuxserver/sonarr-v4-version-4.0.0.240
kope/protokube-1.4.2	kope/protokube-1.4.2	kope/protokube-1.4.2
fluent/fluent-bit-1.9	fluent/fluent-bit-1.9	fluent/fluent-bit-2.0.8
linuxserver/jackett-0.20.1304	linuxserver/jackett-0.20.1992	linuxserver/jackett-0.20.2447
fluxcd/flux-1.25.2	fluxcd/flux-1.25.4	fluxcd/flux-1.25.4
linuxserver/tautulli-2.10.2	linuxserver/tautulli-2.10.4	linuxserver/tautulli-2.11.0
linuxserver/ombi-4.16.12	linuxserver/ombi-4.16.12	linuxserver/ombi-4.22.5
linuxserver/plex-1.27.2	linuxserver/plex-1.29.0	linuxserver/plex-1.30.0
linuxserver/nzbget-version-v21.1	linuxserver/nzbget-version-v21.1	linuxserver/nzbget-version-v21.1
jenkins/jenkins-2.359	jenkins/jenkins-2.371	jenkins/jenkins-2.382
fluent/fluentd-kubernetes-daemonset-v1.14.6-debian-elasticsearch7-1.1	fluent/fluentd-kubernetes-daemonset-v1.15-debian-elasticsearch7-1	fluent/fluentd-kubernetes-daemonset-v1.15-debian-elasticsearch7-1
linuxserver/sabnzbd-3.6.0	linuxserver/sabnzbd-3.6.1	linuxserver/sabnzbd-3.7.0
fluxcd/helm-operator-1.4.2	fluxcd/helm-operator-1.4.4	fluxcd/helm-operator-1.4.4
linuxserver/heimdall-2.4.13	linuxserver/heimdall-2.4.13	linuxserver/heimdall-2.5.3
linuxserver/lidarr-1.0.2	linuxserver/lidarr-1.0.2	linuxserver/lidarr-1.0.2
linuxserver/transmission-version-3.00-r5	linuxserver/transmission-version-3.00-r5	linuxserver/transmission-version-3.00-r6
linuxserver/bazarr-1.1.0	linuxserver/bazarr-1.1.1	linuxserver/bazarr-1.1.3
linuxserver/mariadb-10.5.16	linuxserver/mariadb-10.5.17	linuxserver/mariadb-10.6.9
linuxserver/hydra2-version-v3.0.0	linuxserver/hydra2-version-v3.0.0	linuxserver/hydra2-version-v3.0.0
linuxserver/unifi-controller-7.1.66	linuxserver/unifi-controller-7.2.94	linuxserver/unifi-controller-7.3.76
linuxserver/deluge-2.0.5	linuxserver/deluge-2.1.1	linuxserver/deluge-2.1.1
linuxserver/nextcloud-24.0.2	linuxserver/nextcloud-24.0.5	linuxserver/nextcloud-25.0.2
vaultwarden/server-1.25.0-alpine	vaultwarden/server-1.25.2-alpine	vaultwarden/server-1.27.0-alpine
linuxserver/qbittorrent-4.4.3	linuxserver/qbittorrent-4.4.5	linuxserver/qbittorrent-4.4.5
linuxserver/lazylibrarian-version-52be663d	linuxserver/lazylibrarian-version-dd615ebd	linuxserver/lazylibrarian-version-a803a275
envoyproxy/envoy-v1.20.6	envoyproxy/envoy-v1.23.1	envoyproxy/envoy-v1.23.3
linuxserver/duckdns-version-eb89e848	linuxserver/duckdns-version-7d2a1c41	linuxserver/duckdns-version-13f609b7
coredns/coredns-1.9.3	coredns/coredns-1.9.4	coredns/coredns-1.10.0
linuxserver/syncthing-1.20.3	linuxserver/syncthing-1.20.4	linuxserver/syncthing-1.22.2
linuxserver/calibre-web-0.6.18	linuxserver/calibre-web-0.6.19	linuxserver/calibre-web-0.6.19
linuxserver/duplicati-2.0.6-development	linuxserver/duplicati-2.0.6-development	linuxserver/duplicati-2.0.6-development
envoyproxy/envoy-alpine-v1.20.6	envoyproxy/envoy-alpine-v1.20.7	envoyproxy/envoy-alpine-v1.21.6
linuxserver/nzbhydra2-4.4.0	linuxserver/nzbhydra2-4.4.0	linuxserver/nzbhydra2-4.7.6
linuxserver/rutorrent-version-v3.10	linuxserver/rutorrent-version-v3.10	linuxserver/rutorrent-version-v3.10

openpolicyagent/opa-0.42.2-static	openpolicyagent/opa-0.44.0-static	openpolicyagent/opa-0.46.3-static
linuxserver/ubooquity-2.1.2	linuxserver/ubooquity-2.1.2	linuxserver/ubooquity-2.1.2
jenkins/inbound-agent-4.10-3	jenkins/inbound-agent-4.10-3	jenkins/inbound-agent-4.10-3
fluent/fluentd-v1.15-debian-1	fluent/fluentd-v1.15-debian-1	fluent/fluentd-v1.15-debian-1
jenkins/jnlp-agent-maven-jdk11	jenkins/jnlp-agent-maven-jdk11	jenkins/jnlp-agent-maven-jdk11
linuxserver/jellyfin-10.8.1	linuxserver/jellyfin-10.8.5	linuxserver/jellyfin-10.8.8
linuxserver/hydra-version-0.2.233	linuxserver/hydra-version-0.2.233	linuxserver/hydra-version-0.2.233
linuxserver/muximux-version-418923c8	linuxserver/muximux-version-418923c8	linuxserver/muximux-version-418923c8
linuxserver/mylar-version-94dcfd13	linuxserver/mylar-version-94dcfd13	linuxserver/mylar-version-94dcfd13
linuxserver/headphones-version-83398cb1	linuxserver/headphones-version-83398cb1	linuxserver/headphones-version-a78f38c1
openpolicyagent/gatekeeper-v3.9.0-rc.1	openpolicyagent/gatekeeper-v3.9.0	openpolicyagent/gatekeeper-v3.10.0
linuxserver/swag-1.29.0	linuxserver/swag-1.29.0	linuxserver/swag-1.32.0
linuxserver/tvheadend-version-e3f4f222	linuxserver/tvheadend-version-4741b3c1	linuxserver/tvheadend-version-81c986d5
linuxserver/smokeping-2.7.3	linuxserver/smokeping-2.7.3	linuxserver/smokeping-2.7.3
linuxserver/ddclient-3.9.1	linuxserver/ddclient-3.9.1	linuxserver/ddclient-3.10.0
sqlpad/sqlpad-6	sqlpad/sqlpad-6	sqlpad/sqlpad-6
apache/airflow-2.3.3	apache/airflow-2.3.4	apache/airflow-2.5.0
linuxserver/medusa-1.0.5	linuxserver/medusa-1.0.8	linuxserver/medusa-1.0.9
clearlinux/iperf-3.10.1	clearlinux/iperf-3.10.1	clearlinux/iperf-3.12
testcontainers/ryuk-0.3.3	testcontainers/ryuk-0.3.4	testcontainers/ryuk-0.3.4
linuxserver/nginx-1.20.2	linuxserver/nginx-1.20.2	linuxserver/nginx-1.20.2
linuxserver/bookstack-22.06.2	linuxserver/bookstack-22.09.1	linuxserver/bookstack-22.11.1
linuxserver/resilio-sync-2.7.3	linuxserver/resilio-sync-2.7.3	linuxserver/resilio-sync-2.7.3
kope/dns-controller-1.20.1	kope/dns-controller-1.20.1	kope/dns-controller-1.20.1
linuxserver/minetest-5.5.1	linuxserver/minetest-5.6.1	linuxserver/minetest-5.6.1
linuxserver/grocy-3.3.1	linuxserver/grocy-3.3.1	linuxserver/grocy-3.3.2
linuxserver/code-server-4.5.0	linuxserver/code-server-4.7.1	linuxserver/code-server-4.9.1
linuxserver/beets-1.6.0	linuxserver/beets-1.6.0	linuxserver/beets-1.6.0
apache/couchdb-3.2.2	apache/couchdb-3.2.2	apache/couchdb-3.2.2
linuxserver/wireguard-1.0.20210914	linuxserver/wireguard-1.0.20210914	linuxserver/wireguard-1.0.20210914
linuxserver/freshrss-1.19.2	linuxserver/freshrss-1.19.2	linuxserver/freshrss-1.20.2
linuxserver/calibre-6.0.0	linuxserver/calibre-6.6.0	linuxserver/calibre-6.10.0
authelia/authelia-4.36	authelia/authelia-4.36	authelia/authelia-4.37
linuxserver/emby-4.7.5	linuxserver/emby-4.7.8	linuxserver/emby-4.7.11
linuxserver/sickchill-version-2022.2.20	linuxserver/sickchill-version-2022.9.28	linuxserver/sickchill-version-2022.10.13
linuxserver/piwigo-12.3.0	linuxserver/piwigo-12.3.0	linuxserver/piwigo-13.3.0
photoprism/photoprism-220629-jammy	photoprism/photoprism-220730-jammy	photoprism/photoprism-221118-jammy
pactfoundation/pact-broker-2.102.1.0	pactfoundation/pact-broker-2.104.0.0	pactfoundation/pact-broker-2.105.0.1
linuxserver/domoticz-2022.1.20220630-stable	linuxserver/domoticz-2022.1.20220922-stable	linuxserver/domoticz-2022.2.20221215
stackstorm/st2actionrunner-3.7	stackstorm/st2actionrunner-3.7	stackstorm/st2actionrunner-3.8
silintl/ecs-deploy-3.10.0	silintl/ecs-deploy-3.10.0	silintl/ecs-deploy-3.10.0
linuxserver/cops-1.1.3	linuxserver/cops-1.1.3	linuxserver/cops-1.1.3
linuxserver/htpcmanager-version-15ec6d92	linuxserver/htpcmanager-version-bee309ba	linuxserver/htpcmanager-version-bee309ba
jenkins/slave-4.13.2-1-alpine	jenkins/slave-4.13.3-1-alpine	jenkins/slave-4.13.2-2-alpine
linuxserver/dokuwiki-version-2020-07-29	linuxserver/dokuwiki-version-2022-07-31	linuxserver/dokuwiki-version-2022-07-31a
linuxserver/kodi-headless-190	linuxserver/kodi-headless-190	linuxserver/kodi-headless-190
linuxserver/lychee-4.4.0	linuxserver/lychee-4.4.0	linuxserver/lychee-4.4.0
apache/superset-bd6037ef[...]	apache/superset-3057e427[...]	apache/superset-64496e8c[...]
linuxserver_librespeed-5.2.5	linuxserver_librespeed-5.2.5	linuxserver_librespeed-5.2.5
linuxserver_netbootxyz-0.6.7	linuxserver_netbootxyz-0.6.7	linuxserver_netbootxyz-version-0.6.7
itisfoundation_webserver-master-github-2022-07-13–06-00.e4850d2c[...]	itisfoundation_webserver-master-github-2022-10-04–10-01.6274bb4b[...]	itisfoundation_webserver-master-github-2022-12-21–14-06.b7b180cc[...]
linuxserver_healthchecks-2.2.1	linuxserver_healthchecks-2.3.20220923	linuxserver_healthchecks-2.5.20221230
linuxserver_oscam-version-11711	linuxserver_oscam-version-11713	linuxserver_oscam-version-11718
ensemblorg_ensembl-vep-release_107.0	ensemblorg_ensembl-vep-release_107.0	ensemblorg_ensembl-vep-release_108.2
itisfoundation_director-master-github-2022-07-13–06-00.e4850d2c[...]	itisfoundation_director-master-github-2022-10-04–10-01.6274bb4b[...]	itisfoundation_director-master-github-2022-12-21–12-11.79537769[...]
itisfoundation_storage-master-github-2022-07-13–06-00.e4850d2c[...]	itisfoundation_storage-master-github-2022-10-04–10-01.6274bb4b[...]	itisfoundation_storage-master-github-2022-12-21–14-06.b7b180cc[...]
fluxcd_flux-prerelease-master-2f27d762	fluxcd_flux-prerelease-master-5e59d0c8	fluxcd_flux-prerelease-master-c2317f81
jhipster_jhipster-registry-v7.3.0	jhipster_jhipster-registry-v7.4.0	jhipster_jhipster-registry-v7.4.0
linuxserver_openssh-server-version-8.8_p1-r1	linuxserver_openssh-server-version-8.8_p1-r1	linuxserver_openssh-server-version-9.0_p1-r2
itisfoundation_sidecar-release-github-v1.16.2-2021-10-26–05-40.349c2dc3[...]	itisfoundation_sidecar-release-github-v1.16.2-2021-10-26–05-40.349c2dc3[...]	itisfoundation_sidecar-release-github-v1.16.2-2021-10-26–05-40.349c2dc3[...] 8
openebs_provisioner-localpv-3.3.0-RC2	openebs_provisioner-localpv-3.4.0	openebs_provisioner-localpv-3.4.0
apache_tika-1.28.2	apache_tika-1.28.5	apache_tika-2.6.0.1
linuxserver_mylar3-0.6.8	linuxserver_mylar3-0.6.9	linuxserver_mylar3-0.6.9
apache_nifi-registry-1.16.3	apache_nifi-registry-1.17.0	apache_nifi-registry-1.19.0
itisfoundation_rabbitmq-3.8.0-management	itisfoundation_rabbitmq-3.8.0-management	itisfoundation_rabbitmq-3.11.2-management
apache_nifi-1.16.3	apache_nifi-1.17.0	apache_nifi-1.19.0
linuxserver_thelounge-4.3.1-next	linuxserver_thelounge-4.3.1-next	linuxserver_thelounge-4.3.1-next
clearlinux_redis-7.0.2	clearlinux_redis-7.0.5	clearlinux_redis-7.0.5
vitess_lite-v0.14.0	vitess_lite-v0.14.2	vitess_lite-v0.14.4
linuxserver_webgrabplus-4.2.4	linuxserver_webgrabplus-4.2.4	linuxserver_webgrabplus-5.0.1
treehouses_couchdb-2.3.1	treehouses_couchdb-2.3.1	treehouses_couchdb-2.3.1
kope_kops-controller-1.20.1	kope_kops-controller-1.20.1	kope_kops-controller-1.20.1
linuxserver_znc-1.8.2	linuxserver_znc-1.8.2	linuxserver_znc-1.8.2
jenkins_ssh-agent-4.1.0	jenkins_ssh-agent-4.1.0	jenkins_ssh-agent-4.1.0

linuxserver_snipe-it-6.0.7	linuxserver_snipe-it-6.0.10	linuxserver_snipe-it-6.0.14
linuxserver_embystat-0.2.0	linuxserver_embystat-0.2.0	linuxserver_embystat-0.2.0
itisfoundation_catalog-master-github-2022-07-13–06-00.e4850d2c[...]	itisfoundation_catalog-master-github-2022-10-04–10-01.6274bb4b[...]	itisfoundation_catalog-master-github-2022-12-21–14-06.b7b180cc[...]
ocaml_opam2-staging-ubuntu-20.04-linux-amd64	ocaml_opam2-staging-ubuntu-20.04-linux-amd64	ocaml_opam2-staging-ubuntu-20.04-linux-amd64
openebs_m-apiserver-2.12.2	openebs_m-apiserver-2.12.2	openebs_m-apiserver-2.12.2
stackstorm_st2scheduler-3.7	stackstorm_st2scheduler-3.7	stackstorm_st2scheduler-3.8
vaultwarden_server-1.25.0	vaultwarden_server-1.25.2	vaultwarden_server-1.27.0
clearlinux_python-3.10	clearlinux_python-3.10	clearlinux_python-3.11.0
mautic_mautic-v4	mautic_mautic-v4	mautic_mautic-v4
drud_watchtower-latest	drud_watchtower-latest	drud_watchtower-latest
jenkins_agent-4.13.2-1	jenkins_agent-4.13.2-2	jenkins_agent-4.10-7
fluent_fluentd-v1.14.0-1.0	fluent_fluentd-v1.15.2-1.0	fluent_fluentd-v1.15.3-1.0

References

Docker hub, 2023. https://www.docker.com/products/docker-hub/. (Accessed 15 March 2023).

CSO, 2020. Solarwinds attack explained. https://www.csoonline.com/article/3601508/solarwinds-supply-chain-attack-explained-why-organizations-were-not-prepared.html. (Accessed 15 March 2023).

CSO, 2023. Pytorch suffers supply chain attack via dependency confusion. https://www.csoonline.com/article/3684468/pytorch-suffers-supply-chain-attack-via-dependency-confusion.html. (Accessed 15 March 2023).

Clair, 2023. https://github.com/quay/clair. (Accessed 15 March 2023).

Docker scan, 2023. https://github.com/docker/scan-cli-plugin. (Accessed 15 March 2023).

Grype, 2023. https://github.com/anchore/grype. (Accessed 15 March 2023).

JFrog, Docker Security, 2023. https://jfrog.com/integration/xray-docker-security-scanning/. (Accessed 15 March 2023).

Sysdig secure, 2023. https://sysdig.com/products/secure/. (Accessed 15 March 2023).

Trivy, 2023. https://github.com/aquasecurity/trivy. (Accessed 15 March 2023).

Mills, A., White, J., Legg, P., 2022. OGMA: visualisation for software container security analysis and automated remediation. In: 2022 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 76–81.

Offensive security exploit database. https://www.exploit-db.com/. (Accessed 15 March 2023).

Sultan, S., Ahmad, I., Dimitriou, T., 2019. Container security: issues, challenges, and the road ahead. IEEE Access 7, 52976–52996. https://doi.org/10.1109/ACCESS.2019.2911732.

Shu, R., Gu, X., Enck, W., 2017. A study of security vulnerabilities on docker hub. In: ACM Conference on Data and Application Security and Privacy, pp. 269–280.

Zerouali, A., Mens, T., Robles, G., Gonzalez-Barahona, J.M., 2019. On the relation between outdated docker containers, severity vulnerabilities, and bugs. In: 2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER), pp. 491–501.

Liu, P., Ji, S., Fu, L., Lu, K., Zhang, X., Lee, W.-H., Lu, T., Chen, W., Beyah, R., 2020. Understanding the security risks of docker hub. In: European Symposium on Research in Computer Security, pp. 257–276.

Wist, K., Helsem, M., Gligoroski, D., 2021. Vulnerability analysis of 2500 docker hub images. In: Advances in Security, Networks, and Internet of Things. Springer, pp. 307–327.

Haque, M.U., Babar, M.A., 2022. Well begun is half done: an empirical study of exploitability & impact of base-image vulnerabilities. In: 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE, pp. 1066–1077.

Javed, O., Toor, S., 2021. Understanding the quality of container security vulnerability detection tools. arXiv preprint. arXiv:2101.03844.

Chen, L., Xia, Y., Ma, Z., Zhao, R., Wang, Y., Liu, Y., Sun, W., Xue, Z., 2022. Seaf: a scalable, efficient, and application-independent framework for container security detection. J. Inf. Secur. Appl. 71, 103351.

Dagda, 2021. https://github.com/eliasgranderubio/dagda. (Accessed 15 March 2023).

Anchore-engine, 2023. https://github.com/anchore/anchore-engine. (Accessed 20 August 2023).

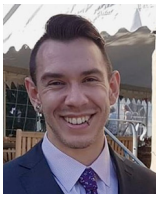
Jacobs, J., Romanosky, S., Adjerid, I., Baker, W., 2020. Improving vulnerability remediation through better exploit prediction. J. Cybersecurity 6 (1), tyaa015.

Jacobs, J., Romanosky, S., Edwards, B., Adjerid, I., Roytman, M., 2021. Exploit prediction scoring system. Digit. Treats Res. Pract. 2 (3), 1–17.

Top 50 products having highest number of cve security vulnerabilities, 2023. https://www.cvedetails.com/top-50-products.php?year=2022. (Accessed 9 July 2023).

AutoHoneyPoC, 2023. https://github.com/ZephrFish/AutoHoneyPoC. (Accessed 15 July 2023).

Ghavamnia, S., Palit, T., Benameur, A., Polychronakis, M., 2020. Confine: automated system call policy generation for container attack surface reduction. In: 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020), pp. 443–458.



Alan Mills is a Lecturer in Cyber Security at University of the West of England (UWE Bristol), having joined on a permanent basis since January 2022. His specialism is in computer and network security, and most recently in containerisation security, vulnerability assessment and hardening. He was the principal engineer for container based solutions that include automated remediation of identified security vulnerabilities. He continues to work within industry securing container-based products and solutions that are sold to multiple government agencies. He has experience of working with hardened containers and compliance requirements for solutions within the DoD. Further details can be found online.

Jonathan White is a Senior Lecturer in Cyber Security at University of the West of England (UWE Bristol). His specialism is in computer and network security, and most recently in IoT systems security including the development of one-class learning approaches for IoT anomaly detection. He joined UWE in January 2020, bringing 23 years of experience from the telecommunications sector (Sonus Networks) from software developer, through to management, and most recently as Director of Product Support, working on some of the world's largest and most complex Critical National Infrastructure. Further details can be found online.

Phil Legg is a Professor of Cyber Security at the University of the West of England (UWE Bristol). His research interests span across cyber security, machine learning, visualisation, and human-computer interaction, to better understand the detection and mitigation of security threats. He has led various research activities as supported by DSTL, NCSC, UKRI, CCAV, CPNI, along with industry and academic collaborators. He has published over 60 academic journals and conference papers across his research interests (citation count: 1694; h-index: 20), with successful research funding of over £2.2M. He is the Programme Leader of the MSc Cyber Security, and Co-Director of the NCSC-supported Academic Centre of Excellence in Cyber Security Education. Before joining UWE in 2015, previous posts have been held at the University of Oxford, Swansea University and Cardiff University. Further details can be found online.