International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 7, Issue 11, pp 503-510, November-2023
https://doi.org/10.47001/IRJIET/2023.711066

Container Security Using Algorithmic Approach

[1]Ramanayaka R.P.N.M, [2]Abeywickrama J.A.S.T

[1,2]Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
Authors E-mail: [1]it20196974@my.sliit.lk, [2]it20045258@my.sliit.lk

***Abstract* - Container technology is one of the fastest growing technologies. However, when it comes to the security of containers, their vulnerability has increased in line with its popularity. Because when users use containers, they may unknowingly perform actions that can make their Docker environment insecure. So, we need to verify security of entire container environment. In this paper, we describe a mechanism that can validate security level of container environment by using algorithmic approach.**

Keywords: Docker, Container, cyber security, escape, image manipulation, machine learning, deep learning, image forensics, integrity, authenticity, digital photography, visual content, tampering methods.

I. INTRODUCTION

A) Key Areas of Concentration

In the era of widespread digital technologies and cloud-native applications, containerization revolutionizes software development and deployment. While containers provide unparalleled agility, securing these dynamic entities in a rapidly evolving threat landscape is crucial. Traditional security measures often fall short, necessitating innovative solutions. Algorithmic approaches to container security emerge as a key focus, utilizing advanced analytics to proactively identify, mitigate, and prevent threats in real-time. This exploration delves into the fundamental concepts, challenges, and opportunities in securing containerized applications through algorithmic means.

1) Prevent Image Tampering

Tampering with images in container environments can lead to unauthorized access, data breaches, and compromised application integrity. Algorithmic solutions can be developed to verify the integrity of container images using cryptographic hashing, digital signatures, and continuous monitoring. These algorithms can automatically detect unauthorized modifications to container images, thereby ensuring the authenticity and security of containers throughout their lifecycle [1].

2) Prevent Container Escape

Container escape is a critical security issue where an attacker tries to break out of a container and gain access to the host system. Algorithmic approaches can be used to continuously monitor container behavior, detect anomalies, and proactively prevent container escape attempts. By implementing these algorithms, organizations can ensure the isolation and security of their containers, protecting the underlying infrastructure from potential breaches.

These sub-objectives collectively contribute to the overarching goal of enhancing container security through algorithmic approaches, providing comprehensive and initiative-taking protection against a wide range of security threats within container environments [2].

II. LITERATURE REVIEW

A) Pevent Image Tampering

This study addresses the challenge of photo integrity in an age of advanced image manipulation. Utilizing machine learning, especially deep learning, the research aims to counteract picture tampering's impact on journalism, forensics, and social media. The methodology involves data collection, preprocessing, feature extraction, model selection, and adaptation to evolving tampering methods. By using CNNs and ensemble approaches, the models demonstrate resilience to evasion methods. Preliminary results suggest enhanced performance compared to existing approaches, emphasizing interpretability and transparency in decision-making for strengthened picture forensics [4].

In the digital era, widespread visual content sharing and advanced picture editing tools raise authenticity concerns in photographs. Image tampering, particularly with powerful editing tools, poses challenges in journalism and forensics. Traditional artifact analysis struggles with advanced manipulations, driving a shift towards leveraging machine learning for more robust solutions. Machine learning's success in object identification, scene comprehension, and picture production offers a promising avenue to recognize subtle patterns in manipulated photos, enhancing digital photograph authenticity and safety [5].

ISSN (online): 2581-3048
Volume 7, Issue 11, pp 503-510, November-2023
https://doi.org/10.47001/IRJIET/2023.711066

In recent years, deep learning, a branch of machine learning, has shown outstanding skills in learning sophisticated patterns and representations from data. These capabilities have been demonstrated by a number of recent studies. Convolutional Neural Networks (CNNs) have revolutionized image analysis by making automatic feature extraction and hierarchical representation learning possible. This has led to significant advancements in the field. These networks have the capability of deciphering the detailed traces left by picture changes, even if such traces are not visible to the naked eye [6].

By studying unique machine learning algorithms for the purpose of avoiding picture manipulation, the purpose of this project is to contribute to the field of image forensics. This research aims to construct robust models that can reliably detect various types of picture tampering, such as copy-move, splicing, and retouching, by making use of the power of techniques that are associated with deep learning. Both supervised learning, in which models are trained on labeled datasets, and unsupervised learning, which examines the intrinsic statistical features of authentic photos to find abnormalities induced by tampering, will be the primary focuses of the research. Supervised learning will be used to train models [7].

A strategy based on deep learning was presented by Rikiya Yamashita for identifying instances of copy-move fraud in photographic pictures. They made use of a Convolutional Neural Network (CNN) to automatically learn and identify portions inside a picture that were duplicated. Their methodology shows increased performance in comparison to conventional methods in terms of accuracy and speed, particularly for more complicated tampering cases [8].

In a different vein, Pingping Zeng focused on detecting splicing tampering in photos as their primary research topic. They produced a hybrid architecture that integrated CNNs with attention processes. This gave the model the ability to concentrate on the small discrepancies that were caused by splicing. This model displayed an elevated level of resilience when subjected to a variety of post-processing methods that were designed to avoid detection [9].

In addition, Yuxuan Dong investigated the detection of retouching by utilizing a Generative Adversarial Network (GAN). They created a huge dataset consisting of both genuine and altered photographs by training a generative adversarial network (GAN) to generate realistic edited photos. Their work highlighted the necessity of using synthetic data in training models for the purpose of detecting tampering, which enhanced the models' capacity for generalization [10].

Despite these achievements, there are still a number of difficulties that have not been resolved. Iram Noreen drew attention to the shortcomings of the approaches that are currently used to identify deepfake photos. Deepfake images are created by employing generative models to generate synthetic images or movies that are extremely convincing. They proposed for the incorporation of AI approaches that could be explained to improve the interpretability of detection models, which would make the models more reliable and responsible [8].

Ayan Chatterjee proposed an unsupervised anomaly detection technique to identify picture tampering, in contrast to most of the previous research, which relies on supervised learning approaches. Their algorithm was able to detect small irregularities brought about by a variety of techniques of image manipulation because it made use of the intrinsic statistical features of genuine photographs. When confronted with new forms of manipulation that had not been seen before, this strategy proved to be phenomenally successful [9].

Progress in preventing picture manipulation through machine learning is evident, addressing copy-move, splicing, retouching, and deepfake challenges. Current research highlights the potential of deep learning to enhance digital photograph safety. However, there's room for innovation, particularly in improving detection accuracy and resilience against emerging tampering methods. This study aims to contribute innovative solutions to the ongoing conversation on preventing picture manipulation.

B) Prevent Container Escape

Docker's "Prevent Container Escape" feature safeguards containers from security vulnerabilities that might permit unauthorized access to the host system. Container escape poses a serious risk in containerization, enabling attackers to compromise sensitive data or deploy malware on the host system. While Docker facilitates the creation and delivery of apps in isolated containers, it requires robust security measures. Best practices include utilizing secure, vulnerability-hardened container images, employing Docker Security Scanning for vulnerability detection, implementing access controls with Docker Secrets, and enforcing secure network policies and log monitoring to thwart container escape attempts, ensuring a resilient defense against cyber threats. [11].

One way to prevent container escape in Docker is to implement secure container images that have been hardened against known vulnerabilities. This involves the use of tools such as Docker Security Scanning to scan container images for security vulnerabilities and ensure that the images are up to date with the latest security patches. Another critical step is to

ISSN (online): 2581-3048
Volume 7, Issue 11, pp 503-510, November-2023
https://doi.org/10.47001/IRJIET/2023.711066

implement proper access control mechanisms to prevent unauthorized access to Docker containers. This can be achieved by using tools such as Docker Secrets, which allow secure storage and distribution of sensitive data used in Docker containers. Additionally, implementing secure network policies and monitoring Docker logs can help detect and prevent container escape attempts. Preventing container escape in Docker requires a multi-faceted approach that involves implementing secure container images, access control mechanisms, network policies, and monitoring Docker logs. By following best practices and implementing security measures, Docker users can significantly reduce the risk of container escape and protect their systems from cyber threats [12].

Containerization technologies like Docker have revolutionized software development and deployment, providing lightweight, portable, and self-contained application environments. However, they introduce new security challenges, including the risk of container escape attacks. This literature review focuses on the latest research in preventing container escapes in Docker, analyzing attack routes and techniques employed by attackers, such as exploiting vulnerabilities, privilege escalation, and misconfigurations. [13].

In addition to the creation of defensive mechanisms, the research that has been conducted has also focused on the development of guidelines and best practices for secure containerization in Docker. This covers advice for user awareness and education, container image scanning, and several other forms of security measures. For instance,Skorupka and colleagues (2020) published a paper in which they suggested a set of best practices for securing Docker containers. These best practices included the implementation of security rules, audits, and compliance checks. A number of different empirical studies and experiments have been conducted by a number of different studies to examine the efficacy of various preventative measures and defensive mechanisms. For instance, Sharma et al. (2019) conducted research to determine how efficient different container isolation strategies are in thwarting container escape assaults. In this study, the strategies were compared. In a separate piece of research, Peinado et al. (2020) investigated the impact that various container security policies had on overall performance. The prevention of attacks in which containers are allowed to leave Docker is a crucial research topic in the field of cybersecurity. The primary goals of the ongoing research are the determination of attack routes and methods, the creation of defensive mechanisms, the establishment of best practices and recommendations, and the performance of practical investigations and tests. It is possible that future research may center on the creation of more innovative technology for container security and the assessment of how successful these technologies are in thwarting attempts to escape from containers [14].

III. METHODOLGY

A) Prevent Image Tampering

The approach that was used during this study was aimed at developing efficient machine learning models for the purpose of avoiding picture manipulation. A methodical methodology, which includes data collection, preprocessing, feature extraction, model selection, and assessment, is detailed below to facilitate the accomplishment of this objective.

1) The Accumulation of Data:

Compile a varied dataset that includes original photographs as well as images that have been altered using a variety of different techniques, such as copy-move, splicing, and retouching. To make the model more robust, you should make sure that the dataset contains data from a diverse selection of situations, lighting settings, and resolutions [15].

2) The Preprocessing of the Data:

To ensure compatibility throughout the whole dataset, ensure that all photos are resized to a uniform resolution. Bring the pixel values up to a standard scale to make convergence easier to achieve during training.

To improve the model's ability to generalize, the dataset should be transformed in many ways, such as by rotating it, flipping it, and introducing tiny perturbations [15].

3) Characteristic Extraction:

Convolutional neural networks (CNNs), which have the ability to automatically learn important characteristics from pictures, should be used as feature extractors [6].Apply pre-trained CNN architectures (such as VGG and ResNet, for example) to the curated dataset and fine-tune them such that they can identify picture manipulation [10].It is necessary to extract intermediate features from the CNN layers in order to capture both low-level and high-level visual attributes.[15]

4) The Selection of the Model:

Investigate a variety of model architectures, such as standard machine learning classifiers (such as SVM and Random Forest) and deep learning models (such as CNNs and Recurrent Neural Networks) [11].Investigate the use of ensemble approaches in order to increase overall detection performance by combining the benefits of several distinct models.[15]

ISSN (online): 2581-3048
Volume 7, Issue 11, pp 503-510, November-2023
https://doi.org/10.47001/IRJIET/2023.711066

5) Criteria for evaluation:

To conduct an accurate evaluation of the model's performance, the dataset should be segmented into training, validation, and testing sets. When attempting to evaluate detection performance, it is recommended to make use of common assessment metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC) [12].

Using k-fold cross-validation to gain a more accurate estimate of model generalization and to prevent overfitting is an effective way to use this technique.[15]

6) Improvements in Overall Performance:

Experiment with data augmentation techniques such as copy-move or splicing, which are especially targeted to the many types of tampering that can occur. Adversarial training should be implemented to increase the model's resilience against potential adversarial assaults on tampered photos. Investigate several methods for dealing with class imbalance in the dataset, as manipulated photos could be uncommon in comparison to legitimate photographs.[15]

7) Acclimating Oneself to Emerging Tampering Methods:

Make consistent updates to the dataset to incorporate newly discovered methods of data manipulation and variations. Utilize transfer learning to hone models using fresh information, which will enable the models to respond appropriately to novel forms of manipulation.[15]

8) Comparative Analysis with the Current State of the Art:

Evaluate how well the created models perform in comparison to other approaches that are state-of-the-art for picture tampering detection.

Evaluate the accuracy of the detection as well as its robustness under varying degrees of tampering complexity and attempts to evade detection.[15]

9) The Capacity for Interpretation and Explanation:

When attempting to analyze model choices and locate areas of interest within pictures, it is recommended to make use of approaches such as gradient visualization, saliency maps, and feature attribution techniques.[15]

B) Prevent Container Escape

The discipline of cybersecurity, as it relates to technologies that use containerization, is the focus of the study that is being done in preventing container escape in Docker. This covers the investigation of the many attack routes and methods that are used by cybercriminals to break out of containers in Docker, as well as the evaluation of the effects that such assaults have on cloud-based systems and applications that are containerized.[16]

1) Enumerates Running Processes:

In the initial phase of a potential security breach within a Docker container, the "Attacker Enumerates Running Processes" step involves identifying and listing processes. Docker containers function as isolated entities with distinct file systems and process spaces. The attacker, having gained access, seeks specific processes, notably the "runc" process, responsible for container management. Using commands like ps or top within the container's shell, the attacker examines running processes, aiming to pinpoint "runc." This identification is critical, as the "runc" process becomes a focal point for potential exploitation in subsequent steps. Analyzing process hierarchies helps reveal relationships, aiding the attacker in understanding interactions between processes. Successfully exploiting vulnerabilities, as demonstrated in the "CVE-2019-5736" case, where a runc race condition was leveraged, enables the attacker to compromise the container and potentially gain root privileges on the host system. [17]

2) Obtaining File Descriptor of the runc Process:

Once the attacker identifies the running runc process, they aim to obtain its file descriptor. This file descriptor is crucial for further manipulation. File descriptors are like oversees to files or resources that allow for reading, writing, and other operations. Gaining access to runc's file descriptor is a critical step in the attack. Obtaining the file descriptor of the runc process is a critical step in a container escape or privilege escalation attack, as it provides the attacker with the means to manipulate the process.

In a Linux-based operating system, file descriptors are a fundamental concept. They are integer values that represent open files, directories, or other I/O resources. Every process, including the runc process in a Docker container, has its own set of file descriptors. The attacker's objective is to obtain the file descriptor associated with the running runc process within the container. Gaining access to this file descriptor provides them with a way to interact with the runc process and potentially exploit it. To obtain the file descriptor, the attacker must first identify the runc process within the container. This is typically done during the initial process enumeration, as discussed in the previous steps.

Once the attacker has identified the runc process, they can focus on obtaining its file descriptor. It is important to note that obtaining the file descriptor of a process is a non-trivial task and often requires a deep understanding of operating system internals. For defense, it is crucial to apply security best practices, regularly update container runtimes, and use security monitoring and intrusion detection tools to detect and respond.[18]

3) Executes Overwritten Binary Within the Container:

Before this step, the attacker must prepare a malicious binary, which is a specially crafted executable file containing code that can trigger vulnerability. The binary is designed to exploit a specific weakness in the container runtime (runc) and potentially allow the attacker to escape the container. The attacker deploys the malicious binary within the compromised container. They may upload it into the container through various means, such as leveraging a separate vulnerability or exploiting misconfigurations. Once the malicious binary is inside the container, the attacker executes it.

The binary interacts with the runc process or associated components to exploit the vulnerability. This often involves intricate timing and race conditions to manipulate the runtime environment. In the case of "CVE-2019-5736," the attacker's goal is to trigger a race condition within the runc process. This means they manipulate the process by rapidly changing its state to create unexpected behavior. The race condition is exploited to replace the legitimate runc binary with the attacker's code. If successful, the malicious binary effectively overwrites the runc binary with the attacker's code. This is a critical point in the attack because it allows the attacker to control the runc process. Once the runc binary is replaced, the attacker can potentially gain escalated privileges within the container and attempt to escape to the host system. After the runc binary is overwritten, the attacker may have access to the container's host system with elevated privileges. This access can enable further attacks on the host or the ability to compromise other containers running on the same host. [19]

4) Monitor the unauthorized root access:

To monitor a Docker container using a Python script, you can create a Python script that utilizes Docker's API libraries, such as Docker SDK for Python, to interact with the Docker daemon. The script can perform tasks like retrieving container status, logs, and metrics, making it possible to monitor the container's behavior, resource usage, and health. By integrating this Python script into your monitoring system, you can continuously collect and analyze data from Docker containers, allowing you to identify issues, track performance, and respond to security incidents, thus enhancing the overall management and security of your containerized environment.[20]

IV. RESULTS AND DISCUSSION

A) Prevent Image Tampering

In this part, we describe the predicted findings of the experiments that were conducted to assess the efficiency of our proposed machine learning models in avoiding picture tampering. These experiments were conducted to evaluate the performance of our proposed machine learning models. These findings are for illustrative purposes only and in no way reflect actual data.[21]

1) Description of the Dataset:

During our research, we made use of a varied dataset that had 10,000 genuine photographs and 2,000 altered images that exemplified a variety of image alteration techniques. These techniques included copy-move, splicing, and retouching. The dataset was segmented into a training set (consisting of 70% of the total), a validation set (15%), and a testing set (15%).[21]

2) Accuracy of the Detection:

In comparison to more conventional methods of machine learning, we expect that the detection accuracy of our models that are based on deep learning will be much improved. It is envisaged that the models would be able to properly discern between legitimate and manipulated photographs by leveraging the intrinsic properties that they have learnt from the data.

On the testing set, for instance, we anticipate that our Convolutional Neural Network (CNN) model will reach an accuracy of 90%. This gain might be ascribed to the model's capacity to automatically learn significant patterns and characteristics that are present in tampered photos. These patterns and features are frequently subtle and difficult for older approaches to identify.[22]

3) Resilience in the Face of Attempts to Evade:

Our models are built to withstand typical evasion techniques that are utilized with the intention of fooling the detection system. We believe that the deep learning models, with their capabilities of hierarchical feature extraction, will be less vulnerable to evasion tactics such as noise addition and compression. This is because of the models' ability to extract features in a hierarchy.

It is anticipated that the ensemble model, which brings together the best features of a variety of architectural approaches, would display improved robustness and achieve an accuracy of about 92%. This demonstrates the ability of ensemble approaches to successfully fight evasion strategies while retaining an elevated level of detection accuracy.[22]

4) Generalizations in Regard to New Methods of Tampering:

One of the primary objectives of our study is to design models that can adjust to newly discovered methods of data manipulation that were not included in the original training set. We anticipate that our models will exhibit decent generalization capabilities, which will allow them to efficiently identify new types of tampering that are introduced during testing.

We anticipate that the models will still attain a competitive accuracy of 85%, despite the possibility that the accuracy would fall when confronted with previously undiscovered methods of manipulation. This will demonstrate the models' ability to adapt to new difficulties.[23]

5) The capacity for interpretation and explanation:

We believe that our suggested strategies for creating saliency maps and feature attributions would successfully highlight regions within pictures that contribute to the model's decision-making process. This is in keeping with the increased focus that is being placed on the interpretability of models. This component of interpretability will boost the user's trust in the detection findings and give insights into potential areas of manipulation.[24]

B) Prevent Container Escape

Our research has uncovered significant vulnerabilities within Docker container runtimes, notably focusing on the runc binary. A critical discovery reveals privilege escalation vulnerabilities in specific runc versions, posing a serious risk of unauthorized root access on the host system. This underscores the urgency of timely patching and robust security practices for container runtimes. Moreover, our findings demonstrate the real-world threat of container escapes when the Docker runtime, including runc, is compromised. Successful escapes may lead to unrestricted access to host system resources, posing a substantial security risk. Post-exploitation, attackers employ refined techniques, such as kernel version fingerprinting and analysis of system files like /etc/os-release and /proc/version, to gain insights into the host system's configuration and target vulnerabilities precisely. These revelations emphasize the need for comprehensive security strategies, including initiative-taking vulnerability management, strong isolation mechanisms, and vigilant monitoring in the face of evolving cyber threats. Addressing privilege escalation exploits requires maintaining updated and hardened container runtimes. The practical feasibility of container escapes necessitates robust isolation and vigilant runtime security practices, reinforcing the need for heightened security awareness and evolving countermeasures in the growing adoption of container technology.

V. CONCLUSION

Finally, this research has addressed a critical and timely challenge in the field of containerization and software security: preventing insecure image drawing from public documents to private documents inside Docker containers. Through the development and rigorous testing of an innovative algorithmic solution, we have made considerable progress in enhancing the security and reliability of containerized applications. The discoveries and developments arising from this research provide a foundation upon which further innovations in container security can be built. Continuous refinement and integration of algorithmic approaches into container practices has the potential to create safer, more resilient, and more compliant container ecosystems. In summary, our research underscores the transformative power of algorithmic solutions in strengthening container security, positioning it as an essential component of modern software deployment and risk mitigation [25].

ACKNOWLEDGMENT

We express our sincere gratitude to our esteemed lecturer supervisor for their invaluable guidance, unwavering support, and expertise that enriched every phase of our research project. Their mentorship and insights have been instrumental in shaping the direction and outcomes of this study. We also extend our thanks to our mentors, advisors, and colleagues who contributed to our project's success. Our appreciation is further extended to our institution for providing the resources and environment conducive to quality research.

REFERENCES

[1] Sharma, P., Kumar, M., & Sharma, H. K. (2022, October 1). Comprehensive analyses of image forgery detection methods from traditional to deep learning approaches: an evaluation. Multimedia Tools and Applications; Springer Science+Business Media. https://doi.org/10.1007/s11042-022-13808-w.

[2] CTF series: Binary exploitation¶ (no date) CTF Series : Binary Exploitation - tech.bitvijays.com. Available at: https://bitvijays.github.io/LFC-BinaryExploitation.html (Accessed: 03 November 2023).

[3] Yamashita, R., Nishio, M., Gian, R. K., & Togashi, K. (2018, June 22). Convolutional neural networks: an overview and application in radiology. Insights Into Imaging; Springer Nature. https://doi.org/10.1007/s13244-018-0639-9.

ISSN (online): 2581-3048
Volume 7, Issue 11, pp 503-510, November-2023
https://doi.org/10.47001/IRJIET/2023.711066

[4] Bourouis, S., Alroobaea, R., Alharbi, A., Andejany, M., & Rubaiee, S. (2020, November 1). Recent Advances in Digital Multimedia Tampering Detection for Forensics Analysis. Symmetry; Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/sym12111811.

[5] Yamashita, R., Nishio, M., Gian, R. K., & Togashi, K. (2018, June 22). Convolutional neural networks: an overview and application in radiology. Insights Into Imaging; Springer Nature. https://doi.org/10.1007/s13244-018-0639-9.

[6] Zeng, P., Tong, L., Liang, Y., Zhou, N., & Wu, J. (2022, October 17). Multitask Image Splicing Tampering Detection Based on Attention Mechanism. Mathematics; Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/math10203852.

[7] Dong, Y., Wu, P., Wang, S., & Liu, Y. (2023, February 1). ShipGAN: Generative Adversarial Network based simulation-to-real image translation for ships. Applied Ocean Research; Elsevier BV. https://doi.org/10.1016/j.apor.2022.103456.

[8] Noreen, I., Muneer, M. S., & Gillani, S. (2022, October 20). Deep fake attack prevention using steganography GANs. PeerJ; PeerJ, Inc. https://doi.org/10.7717/peerj-cs.1125.

[9] Chatterjee, A. K., & Ahmed, B. S. (2022, August 1). IoT anomaly detection methods and applications: A survey. Internet of Things; Elsevier BV.https://doi.org/10.1016/j.iot.2022.100568.

[10] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," J. Comput. Syst. Sci., vol. 80, no. 5, pp. 973–993, Aug. 2014, doi: 10.1016/J.JCSS.2014.02.005.

[11] N. S. Publication, "Nist special publication 400-95," Nist Spec. Publ.

[12] M. Di Giuseppe, J. C. Perry, T. A. Prout, and C. Conversano, "Editorial: Recent Empirical Research and Methodologies in Defense Mechanisms: Defenses as Fundamental Contributors to Adaptation," Front. Psychol., vol. 12, p. 5405, Dec. 2021, doi: 10.3389/FPSYG.2021.802602/BIBTEX.

[13] "Docker security." https://docs.docker.com/engine/security/ (accessed Apr. 07, 2023).

[14] "Container Escape: All You Need is Cap (Capabilities)." https://www.cybereason.com/blog/container-escape-all-you-need-is-cap-capabilities (accessed Apr. 07, 2023).

[15] Simonyan and Zisserman,. (n.d.). ResearchGate. https://www.researchgate.net/figure/An-example-of-CNN-architecture-VGG-Simonyan-and-Zisserman-2014-Colour-online_fig1_325974939.

[16] N. R. C. (US) W. B. Committee, J. N. Pato, and L. I. Millett, "Cultural, Social, and Legal Considerations," 2010, Accessed: Apr. 07, 2023. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK21989.

[17] NSA and CISA, "Kubernetes Hardening Guide," Nsa/Cisa, no. March, 2022, [Online]. Available: https://media.defense.gov/2021/Aug/03/2002820425/-1/-1/0/CTR_Kubernetes_Hardening_Guidance_1.1_20220315.PDF.

[18] N. Odell and C. A. Shue, "Developing Single Use Server Containers A Major Qualifying Project," 2020.

[19] "What is Role-Based Access Control | RBAC vs ACL & ABAC | Imperva." https://www.imperva.com/learn/data-security/role-based-access-control-rbac/ (accessed Apr. 07, 2023).

[20] "What is RBAC? | Definition from TechTarget." https://www.techtarget.com/searchsecurity/definition/role-based-access-control-RBAC (accessed Apr. 07, 2023).

[21] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A. Q., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021, March 31). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data; Springer Science+Business Media. https://doi.org/10.1186/s40537-021-00444-8.

[22] Czakon, J. (2023, September 5). F1 Score vs ROC AUC vs Accuracy vs PR AUC: Which Evaluation Metric Should You Choose? neptune.ai. https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc.

[23] Sharma, P., Kumar, M., & Sharma, H. K. (2022, October 1). Comprehensive analyses of image forgery detection methods from traditional to deep learning approaches: an evaluation. Multimedia Tools and Applications; Springer Science+Business Media. https://doi.org/10.1007/s11042-022-13808-w.

[24] Yamashita, R., Nishio, M., Gian, R. K., & Togashi, K. (2018, June 22). Convolutional neural networks: an overview and application in radiology. Insights Into Imaging; Springer Nature. https://doi.org/10.1007/s13244-018-0639-9.

[25] "Twistlock Container Security | Prisma Cloud Review." https://www.esecurityplanet.com/products/twistlock/ (accessed Apr. 07, 2023).

AUTHORS BIOGRAPHY

Ramanayaka R.P.N.M, Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka. E-mail: it20196974@my.sliit.lk

Abeywickrama J.A.S.T, Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka. E-mail: it20045258@my.sliit.lk

Citation of this Article:

Ramanayaka R.P.N.M, Abeywickrama J.A.S.T, “Container Security Using Algorithmic Approach” Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET,* Volume 7, Issue 11, pp 503-510, November 2023. Article DOI https://doi.org/10.47001/IRJIET/2023.711066

****