Research Project

Vulnerability Analysis and Enhancement of Security in Drupal Containers on AWS: A Multi-tool Approach.

Chittranjan Kumar Sharma

Research Project
Supervisor : Sean Heeney
National College of Ireland
IFSC, Dublin

National College of Ireland

National College of Ireland

Project Submission Sheet – 2022/2023

Student Name: Chittranjan Kumar Sharma

Student ID: x21224960……

Programme: Msc. Cloud Computing…… **Year:** 2023……

Module: Research Project……

Lecturer: Sean Heeney ……

Submission Due Date: 14th August 2023……

Project Title: Vulnerability Analysis and Enhancement of Security in Drupal Containers on AWS: A multi-tool Approach…

Word Count: 4278……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: Chittranjan Kumar Sharma……

Date: 13th August 2023……

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Abstract

In today's technological landscape, Docker containers running on Aws cloud have revolutionized the way applications deployed on cloud by offering scalable, lightweight, and efficient solutions. However, with popularity also comes critical security concerns. This research delves into vulnerabilities and security concerns present in Docker containers, particularly Drupal CMS containers running on Amazon Aws cloud. The research aims to fulfil the gap in existing literature review by proposing the Vulnerability Analyzer and Securing Container Framework. The model was designed and implemented to identify vulnerabilities and subsequently increasing the security of Drupal containers running on Aws cloud. Aqua Trivy, Anchore Engine, Chef, Apparmor, Synk, Selinux played important role in this research and proved that they are capable in reducing vulnerabilities of containers. Therefore, a multi tool approach proved to be successful in finding vulnerabilities in Drupal container security and eventually able to mitigate them.

Keywords :Docker, Containers, Vulnerability, Analysis, Drupal CMS, AWS, Amazon Web Services, Security, Framework, Aqua Trivy, Anchore Engine, Apparmor, SELinux, Synk, Multi-tool, Security Enhancement, Container Hardening.

Contents

List of Figures

Introduction

The growing adoption of containerization technologies, particularly Docker, has positioned itself as the preferred choice in cloud computing for development, deployment, and management of applications. In the past, virtualization was the innovation designed to address the issue of underutilized hardware resources. Virtual machines allowed users to leverage different operating systems on a single machine. However, as highlighted bySultan et al. (2019), containers have emerged as a compelling alternative to virtual machines. This is due to their basis in operating system-level virtualization, which makes them lightweight, easy to deploy, and capable of version control, marking them as a superior option over VMs. Nevertheless, containers continually face significant security challenges, and the underlying infrastructure remains a pivotal concern when deploying applications on Docker. In this research, Drupal CMS is utilized for testing and evaluation. Regarding security threats and vulnerabilities in containers,Wong et al. (2023) has pointed out numerous gaps in current literature. Consequently, this research is centred on the analysis and assessment of vulnerabilities in Docker images, specifically emphasizing Drupal CMS images. It will also evaluate various security tools to determine their effectiveness in mitigating the identified vulnerabilities, aiming to bridge the existing gaps in the state-of-the-art.

1.1 Background and Motivation

Docker has become one of the best containerization platform as a large range of applications including IoT and fog and edge computing are using it. While a Dockerized application like Drupal is made in secure environment, it could be sensitive to a lot of security challenges when addressed. This arises a need to check whether the tools are capable in mitigating these issues like Drupal Container running on Aws cloud are secure or not. Given prevalent use of containers, they are also targeted by attackers in cyber security that intend to hamper its security. Moreover, as many organizations are shifting to cloud platforms like Amazon Aws, and when they are orchestrated with containers, the associated risk with them increases. So, the motivation behind this research is an exclusive analysis of Drupal CMS images running on containers on Aws cloud by facilitating a variety of security tools in reducing the vulnerabilities. Additionally, this necessitates a robust mechanism to identify and mitigate such vulnerabilities, ensuring the secure operation of Drupal CMS inside Docker containers is the primary reason to conduct this research. Although, there has been a lot of research existing on Docker security but when it comes to Drupal CMS there exists gap in research in using multiple security tools concentrated on Drupal. Therefore, it arises a need for deep

examination of Containers' vulnerabilities and their countermeasures simultaneously.

1.2 Research Question

How well can the vulnerabilities be found in Docker Drupal Container running on Aws cloud and what are the comparative improvements be made in security of it using different tools like Trivy, chef, Anchore Engine, AppArmor, and Selinux?

1.3 Research Objectives

Below are the primary objectives for this research:

1. Establish a testing environment for this research framework using Aws EC2 Linux instance with installing Docker and Drupal.

2. Doing comprehensive comparison and contrast of vulnerabilities analysis by fetching Drupal CMS images n Docker using security tools: Aqua Trivy, Anchore Engine, Chef, AppArmor, and Selinux.

3. Although variety of approaches approaches have already been studied in the past, but most of them have gone obsolete. So, for creating a strong defence mechanism, detection of vulnerabilities with appropriate tools is much needed.

4. Furthermore, evaluation of security with SELinux by changing mode from "Permissive" to "Enforcing" on container will be done. Additionally, Drupal container will be configured with Chef vulnerabilities scanner to find threats.

5. Another objective is to provide recommendations based on the findings to enhance security of Drupal containers running on Aws cloud.

By following this approach or planning, this report will aim to answer the proposed research question by providing imminent contributions to the field of Docker security. This research is structured into following chapters, starting with comprehensive Literature review that requires the background understanding of the tools like Docker, Drupal CMS, Anchore Engine, Aqua Trivy, Chef, AppArmor, and SELinux used in this study. Following this, the research methods and specifications will be outlined with details of design, tools and techniques used in evaluation. Moving ahead, the section will discuss the results and concludes

with the summary of entire research and future work recommendations.

Related Work

The primary focus of This literature review is to establish a foundation for this research in containers by comprehensively reviewing the current research present in Docker security and the security loops present in it. The aim is to identify vulnerabilities in the current scenario and developing remedies to heal these issues in containers. Moreover, this review will analyze the efficiency of security tools for Docker containers with primary focus on Anchore Engine, SELinux, Aqua Trivy, chef, and Apparmor. Next sub sections will go deep in State-of-art by providing contrast study of existing literature.

2.1 Docker Containers

According to the author, MerkelMerkel et al. (2014), the architecture of Docker offers a lightweight platform for deploying applications on containers with focus on speed. The reason behind this adoption of docker in real world is the vast toolkit and documentation of it. However, wongWong et al. (2023) provided a deep analysis of threats and vulnerabilities present in containers. His research goes deep in study of attacks against container-based platforms like Docker with highlighting the need to identifying vulnerability and providing solution to improve container security. While Merkel, proposed a framework STRIDE for threat modelling and examination of security of containers. STRIDE is a comprehensive security framework therefore both the authors call for enhancing security in containers. So, it was the reason to choose Drupal CMS running on Docker container to analyze its vulnerabilities and trying to improve its security.The author'sGarg & Garg (2019) findings include the CI/CD pipeline using docker containers to achieve the maximum out of docker but the main challenge is to keep it secure.

2.2 Vulnerabilities in Docker Images

As per the findings of GummarajuGummaraju et al. (2015), more than 30% of the official Docker images are on high risk so to avoid the potential vulnerabilities when an attacker could get unauthorized access to host system, it becomes necessary to find the vulnerabilities associated with containers as quickly as possible. According to author, there could be many reasons for these vulnerabilities like operating system flaws, misconfigurations, and dependencies. Therefore, to resolve these issues, this research is focused on identifying risks

in Docker images by using multiple tools like Anchore Engine, Selinux, Chef, Trivy, and Apparmor.

2.3 Vulnerability Scanning

Next, the work of javedJaved & Toor (2021) was examined carefully, the author uses three different tools to analyze Docker Hub. These are Clair, Anchore Engine, and Microscanner, according to the author's findings Anchore Engine has the best detecting capabilities. However, the tools are not absolute as they have several shortcomings in identifying OS package vulnerabilities, these could be the issue with dependencies. These observations are important for this research's' point of view as it will make use of these tools to identify vulnerabilities in containers.

2.4 Drupal CMS and Docker

Moving ahead, the next author patelPatel et al. (2011), conducted comparative analysis of three famous Content Management Systems(CMS), namely Joomla, Drupal, and Wordpress. The author discovered that Drupal to be the option go for in terms of performance better than others. So this was the reason for choosing Drupal for this research. The better performance was due to its adaptability, user-interactive, and capability for growth with Docker Hub. Additionally, drupal can be subjected to security threats, therefore chosen to conduct research to find vulnerabilities.

2.5 Mitigation Tools

The authorSengupta et al. (2021) suggests that instead of Docker's scalabillity and lightweigh nature they are vulnerable to vulnerabilities. Findings of this paper also highlights the concern that current methodologies are incapable in resolving container's security issues. Thus, the need for new framework is required.

Methodology

This section will decide the methodology of this research in improving security of containers especially Docker, understanding the need of research it requires a well-structured approach. Despite the available research, some of them are outdated or have become ineffective because security is a real time process that should be keep up to date to have secured application.

This research proposes a finite model of finding vulnerabilities and an enhanced defense framework for improving Drupal containers. The model is created with custom-based rules and appropriate permissions by the followings:

- Starting the project by launching Aws Linux EC2 cloud instance for base operating system,
- Analyzing Vulnerabilities of Docker images to protect the Drupal Docker containers from any exploitation,
- Testing of security tools like Aqua Trivy, Anchore Engine, Selinux, AppArmor, Chef vulnerability scanner, and Synk.
- For having a utilized system, metrics will be collected by noticing cpu utilization on Aws cloud. As by doing so the balance between system security and performance can be maintained.

3.1 Research Tools and Techniques:

Following are the tools and techniques used in this research:

1. **Amazon Aws EC2 Linux:**A Linux EC2 t.2micro free tier eligible was used to deploy, test, monitor for this research installation of Docker containers. The reason for choosing it lies in its scalable nature. Amazon Aws is the foundation of this whole project as the complete implementation is executed using cloud resources.
2. **Docker:**Docker[1] is the epicenter of this research used for running containers carrying Drupal CMS images.
3. **Security Toolkit:**A group of security tools including Anchore Engine, Aqua Trivy, SELinux, AppArmor, Snyk[2], and Chef were employed at various stages of the research to analyze and enhance security of containers. The Anchore Engine analyzes the Docker images by identifying vulnerabilities and providing a detailed report on the same. The Anchore Engine performs a deep image inspection and policy-based compliance checks, contributing to a more secure Docker environment. Furthuremore, SELinux[3] policies are used to control access permissions of Drupal CMS images running on Docker containers. SELinux operates on the principle of 'least privilege' where applications and processes are given the minimum permissions they need to function, reducing the potential damage from vulnerabilities.
4. **PostgreSQl:**To manage assessments of vulnerability from Anchore Engine, a relational database PostgreSQL is used.
5. **Github:**Github is used to fetch docker images and datasets for vulnerabilities testing, from its repositories for testing of this research.

[1] `https://docs.docker.com/search/?q=security`
[2] `https://snyk.io/`
[3] `https://linuxconfig.org/how-to-check-selinux-operational-mode`

3.2 Proposed Architecture

Figure below describes the architecture of the research:

1. **Environment setup:**Amazon EC2 linux and ubuntu platform served as the base platform for this research, where Docker and Drupal CMS images are installed. Docker has chosen for Drupal CMS images because Docker simplifies the managing process as it is easy to install dependencies for Drupal CMS.

2. **Vulnerability scanning:**Aqua Trivy , Anchore Engine[4], Synk, and AppArmor are integrated for continuously assessment of containers for finding vulnerabilities.

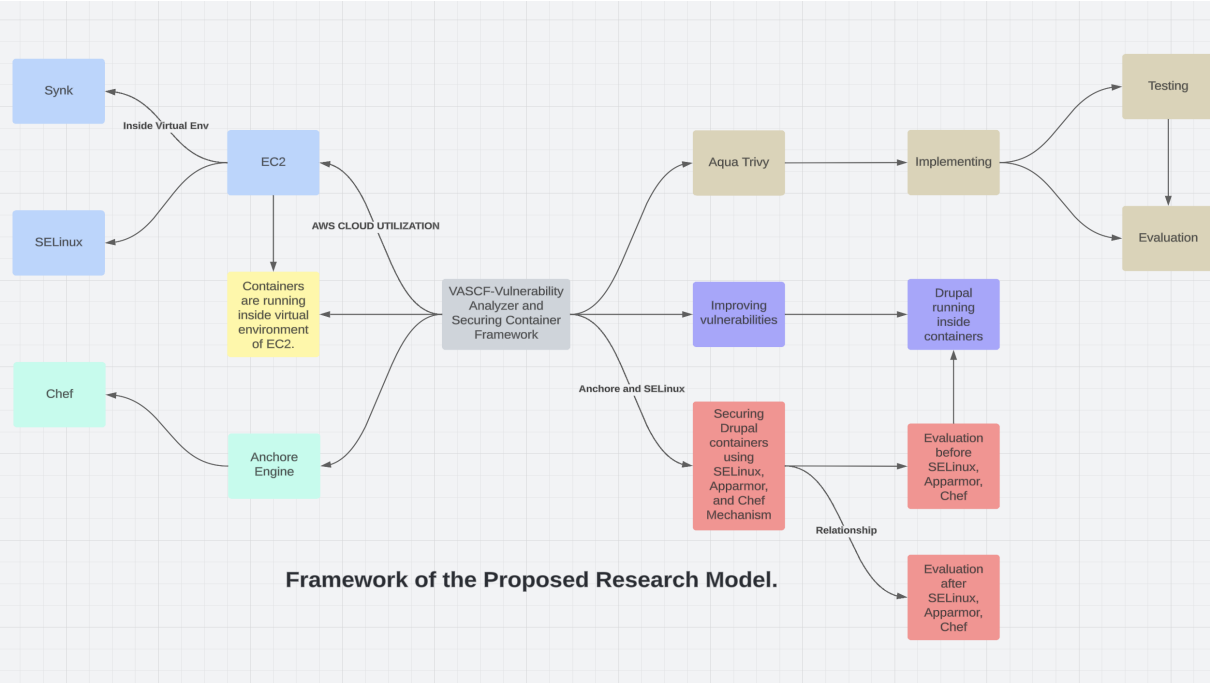


Figure 3.1: Vulnerability Analyzer and Securing Container Framework Research Model

3. **Database Integration:**PostgresSQL is used to store data from Anchore Engine.

4. **Enhancement Layer:**After finding the vulnerabilities above, Selinux, AppArmor, and Chef were utilized to enchance system security. Particularly, Selinux was used in enforced mode to increase security of Drupal CMS. Also, AppArmor enhanced container security by restricting the unauthorized access from system. Additionally, Chef was used to run Drupal container with chef configuration to enhance container security.

5. **Monitoring:**Synk was configured for monitoring the real-time container running Drupal images. In addition to that, throughout the project monitoring of EC2 was done by keep checking different stages of cpu utilization to notice any changes before and after security enhancement of containers.

[4]`https://hub.docker.com/r/anchore/engine-cli`

3.3 Research Strategy

As it can be seen in the Figure showing the methodology for this research, the architecture illustrates the development, testing, and deployment phrases of the project. First step is to initiate a cloud setup, for that Aws EC2 instance is created with Linux configuration after that it is accessed from ssh using pc's terminal and docker is installed on the Linux instance. Secondly, Drupal images are pulled from github and installed in container running on Linux, following it the vulnerability testing phase begins, for that first the dependencies and pre-requisites are installed using cli for the setup of Anchore Engine, Trivy, Selinux, Apparmor, Chef, and Synk. Also, Figure shows that PostgresSQL has been utilized and configured with port of the EC2 instance running Anchore Engine to store the data and findings discovered by Anchore Engine.

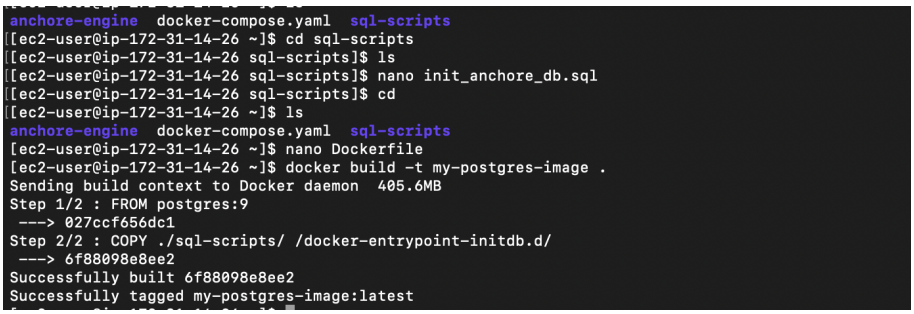


Figure 3.2: PostgresSQL configured to Anchore Engine

In third step, the focus will be on enhancing security of the Drupal container by performing testing and evaluation of tools used – anchore engine, apparmor, chef, and synk. The implementation of SELinux enforcement on Drupal CMS images running on Docker containers, is done to do the isolation of containers by restricting access to system resources and preventing unauthorized actions. Hence, improving system security. So, basically Drupal containers are tested before implementing security policies and vulnerabilities are identified and the evaluation is done again with improved security to measure the reduction in vulnerabilities of Drupal Container.

Design Specifications

As the vulnerability and securing of container Framework can be seen in the Figure. showing all the stages and elements used in the implementation of this research. The main responsibility any defence system is the capability to provide protection to containers by defining certain set of rules and policies and provide continuous monitoring. Thus, the de-

sign specifications for this research structures framework for this implementation to achieve the objectives defined in the research question are mentioned below:

4.1 System Environment:

1. Operating System: Amazon EC2 Linux instance has been used due to its ability to address security concerns when assessing docker vulnerabilities as in this research.

2. Docker 20.10.21 version has been integrated on Cloud container. It is the containerized platform that uses its pull and push commands to enable the creation of containers will be used for deployment and management of containerized Drupal application in this research.

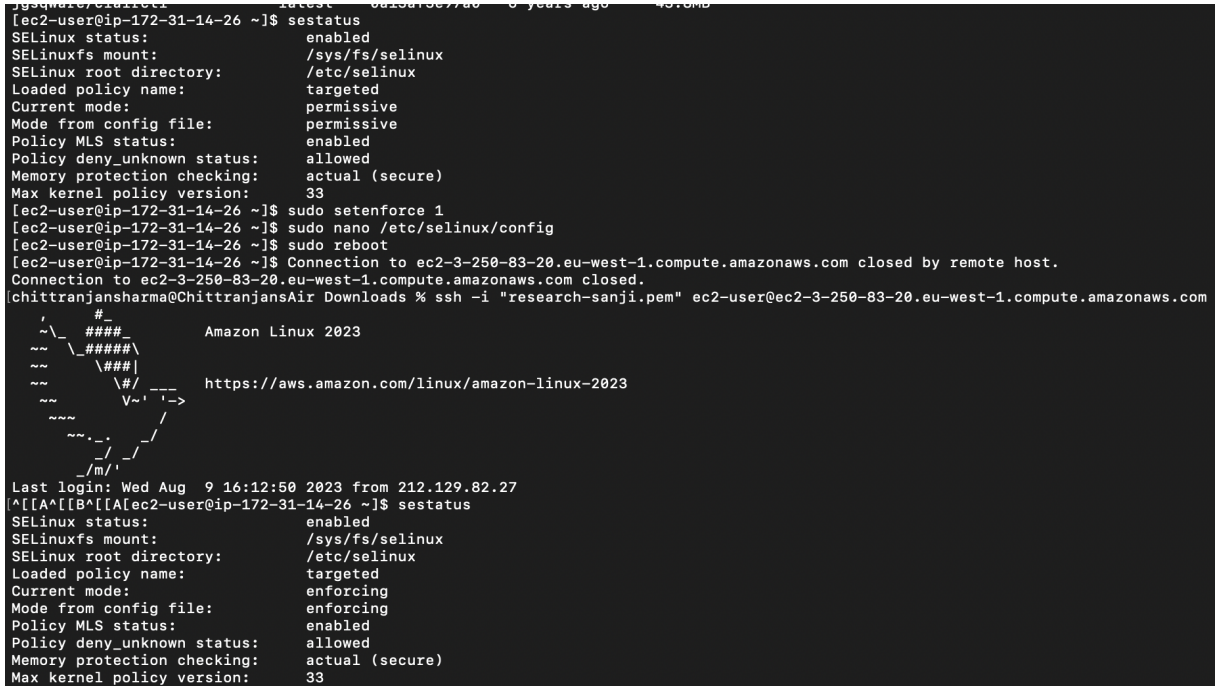


Figure 4.1: SELinux mode changed from permissive to Enabled.

3. Latest version of Drupal CMS running in container-76ca9ff9c0a6 is used for this research.

4. Configurations have been made with each tool- Trivy[1], Anchore Engine, Apparmor, Chef, and Synk by installing dependencies and upgrading them to make it compatible with the Docker container.

[1] https://aquasecurity.github.io/trivy/v0.44/

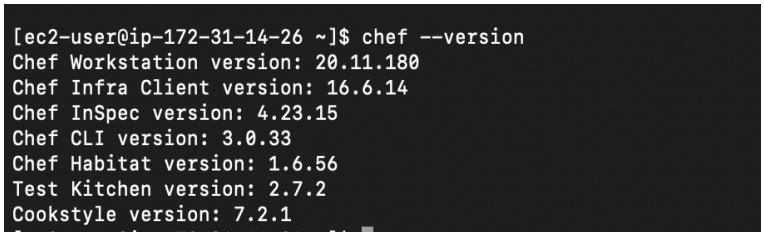


Figure 4.2: Chef scanner version

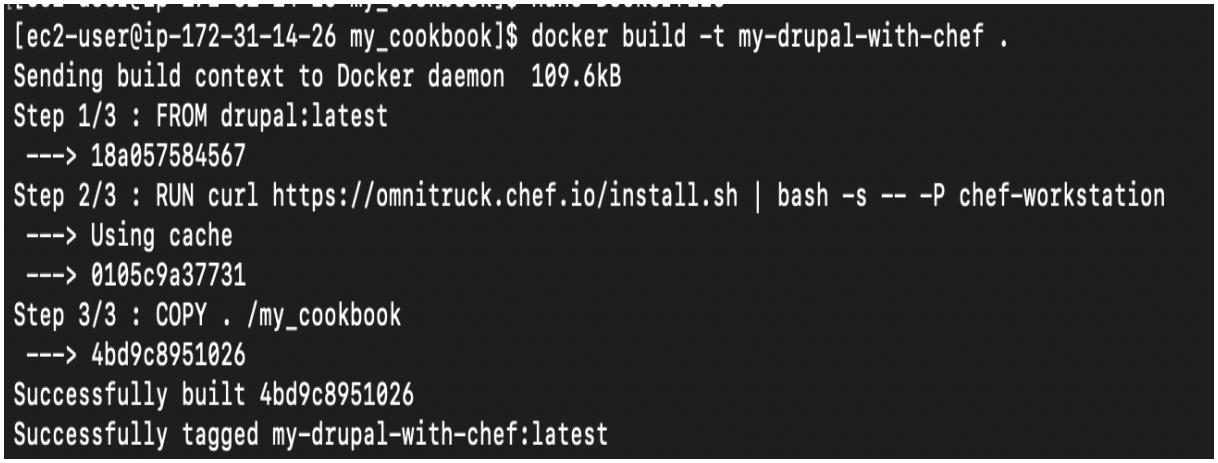


Figure 4.3: Chef vulnerability scanner configuration

4.2 Security Enhancement:

1. Scanning Tools: Integration of Aqua Trivy, Anchore Engine and Chef has been done to achieve the objectives defined in research question to find vulnerabilities in container and to find how efficient they are in evaluating container security.

2. SELinux: The implementation of selinux allows to explore how can drupal container be made secure when Selinux is changed from permissive mode to enforced.

3. Apparmor: This tool is utilized to find the vulnerabilities and securing the container by limiting the least privileges to access container.

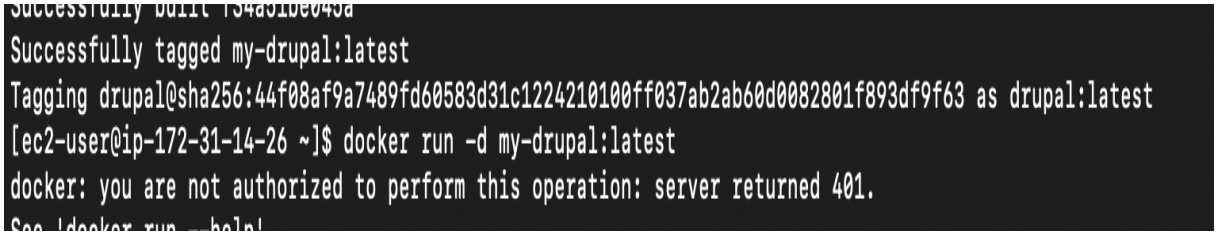


Figure 4.4: User restricted by Apparmor

4.3 Evaluation Mechanism

1. Monitoring Tools: Synk is tested in this research to serve the goal of providing continuous vulnerability asseeement as it provides detailed report of threat CVE's by segmenting them into categories as low, high, severe, and critical.

2. Performance metrics: Given that any security implementation impacts system performance, tools like Gfana to provide essential insights have been tried out but due to their limitation in free version were not included.

3. Comparison: To check the improvement in container security first the vulnerabilities are found out by Synk and Trivy without Apparmor, Chef, and Selinux enforcement. Secondly, vulnerabilities are found out after integration of Apparmor, Chef[2], and Selinux that can be seen in Figure chef.

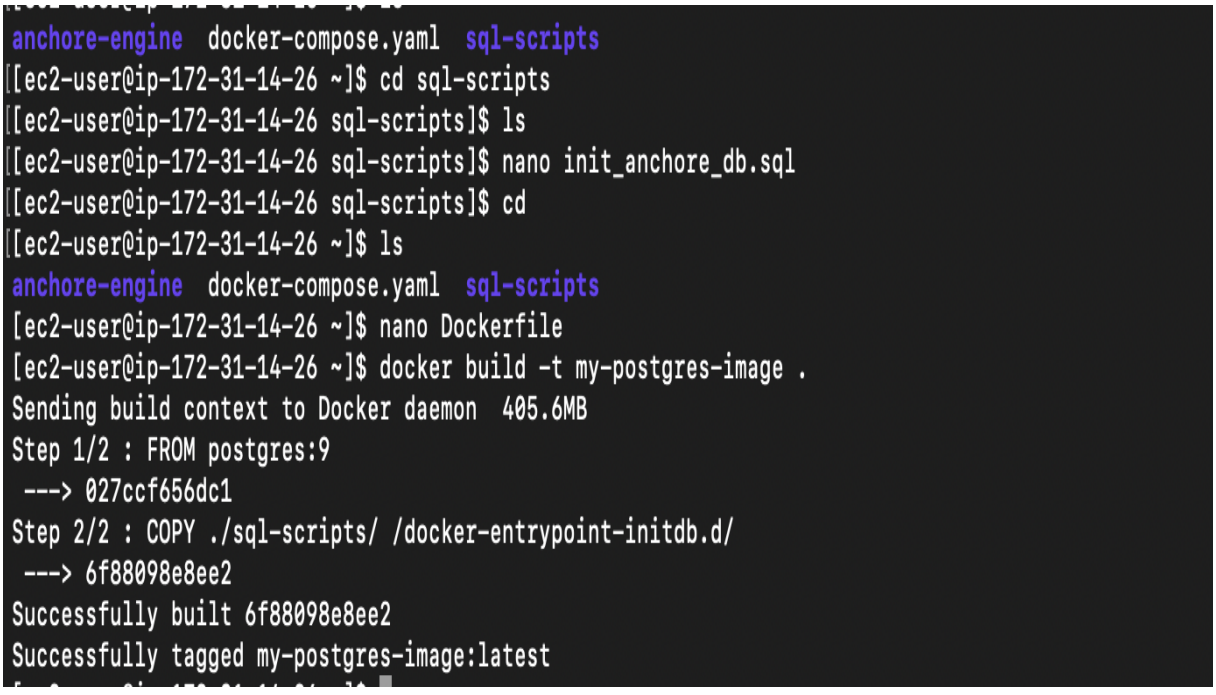


Figure 4.5: PostgresSQL configured to Anchore Engine

[2]https://github.com/sous-chefs/docker

Implementation

The research's Vulnerability Analyzer and Securing Container framework is a model to find vulnerabilities in the Containers especially Drupal CMS running containers on cloud by using several tools and analyzing of security tools to find how efficient are they in order to secure Drupal container. As the experimental study of Drupal container is based on two experiments so the implementation for this research will explains implementation of both the case studies:

- Vulnerability analysis of Drupal Container running on Aws
- Securing Drupal by reducing vulnerabilities.

As there are many Linux techs available for instance Anchore Engine, Aqua Trivy, chef, Apparmor, Selinux, Seccomp and Synk to provide a strong secure system for Docker containers. An EC2 Linux is created with expandable volume of GB after that docker is installed on top of it using docker compose following that Drupal images are pulled from Github using docker pull and stored in drupal container. After that docker compose yaml file is updated with necessary dependencies, moving further Anchore Engine is installed along with anchore cli and anchore image is pulled from github and engine is started to check the vulnerabilities. To store the vulnerabilities postgres is configured to the same port of EC2 instance where Anchore Engine is running. Additionally, many tools were tried like Aqua Trivy, Modsecurity, CloudFlare, Modevasive, Sysdig[1], Neuvector, Chef, Ansible, veeam, Prometheus[2], Automax, Synk, Clair, Falco, JFrog Xray, Black Duck, Quay, and Kube-hunter but due to license restriction the research is proceed with Aqua Trivy, Anchore Engine, Apparmor, Synk, Chef, and SELinux to achieve the evaluation of first case study – Vulnerability finding in Drupal containers and for second case study to secure containers by reducing vulnerabilities. Trivy generated a comprehensive vulnerabilities report with specifying low, high, and critical threats to Drupal container. After that Drupal is configured with chef scanner and a custom Drupal chef container is built, and finally, it is tested again with Apparmor and Selinux and it produces the comprehensive vulnerabilities report with reduced vulnerabilities.

[1] `https://github.com/draios/sysdig/releases`
[2] `https://github.com/prometheus/prometheus/releases`

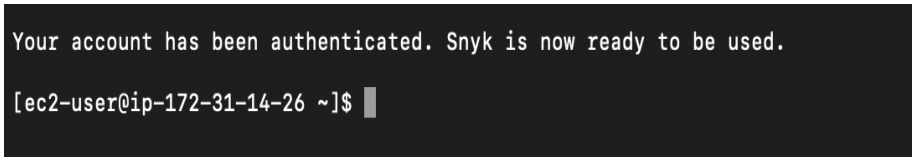


Figure 5.1: Synk Authentication.

Also, Synk was installed to provide continuous monitoring of container can be seen in Figure synk. Furthermore, scan was performed with configuring the container with Apparmor and in results it denied the permissions to access the container that can be seen in Figure apparmor which is expected as the container was configured with apparmor to improve security.

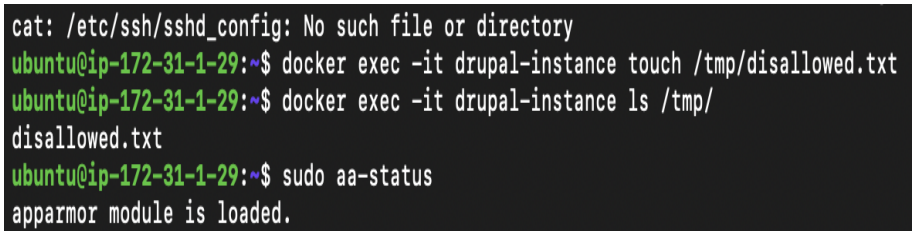


Figure 5.2: Permission denied by Apparmor.

Figure vulner represents the vulnerability report before hardening the Drupal Container and Figure vulner represents the vulnerability report after hardening or securing the Drupal container with reduced vulnerability and better security. Figure shows the containers and images on Docker running on Aws Cloud.

Experiments/Evaluation

This section shows the industrious work done in putting the proposed framework into practicality by performing following experiments and case studies. The docker images in Figure can be seen with "docker ps command" are all obtained from Docker repositories by using "docker pull command". The reason for choosing the images from there is to avoid testing unofficial images and to have official images. After installing the images, first these were analyzed for vulnerabilities. The vulnerabilities status of Drupal container images is categorized into low, high, and medium risk. Subsequently, security enhancement of containers is performed. The experiments are divided into two case studies to provide more clarity:

- Case Study 1 – Vulnerability Analysis of Drupal Container running on Aws.

- Case Study 2- Securing Drupal by Reducing Vulnerabilities.

Let's discuss these experiments in more details below:

6.1 Case Study/ Experiment -1

For this first experiment the objective was to identify existing vulnerabilities in the drupal container using the tools Aqua Trivy and Anchore engine that been mentioned in the previous implementation section.

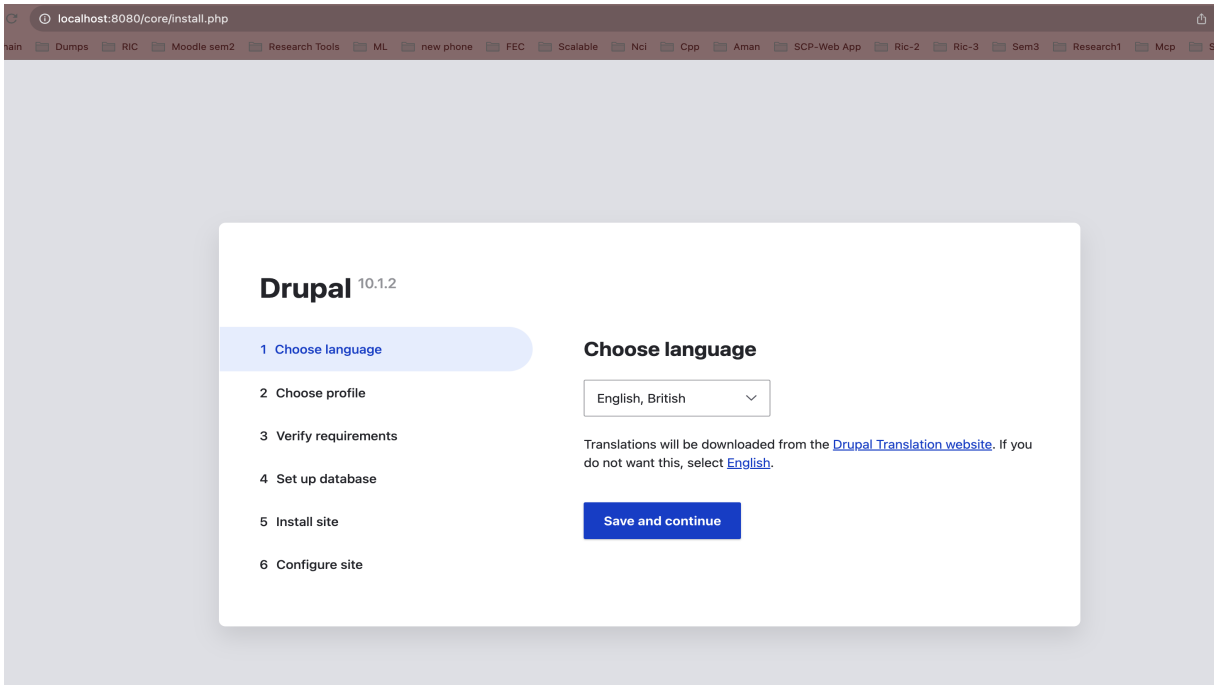


Figure 6.1: Drupal CMS container running on Aws EC2.

For this experiment Drupal image was fetched from Github and loaded into the container running on EC2 linux to discover potential threats that can be seen in Figure drupal. The vulnerabilities are classified as Low, medium, high, and critical in the reportBinnie & McCune (2021). Figure vulner shows the identified vulnerabilities before hardening the Drupal container.

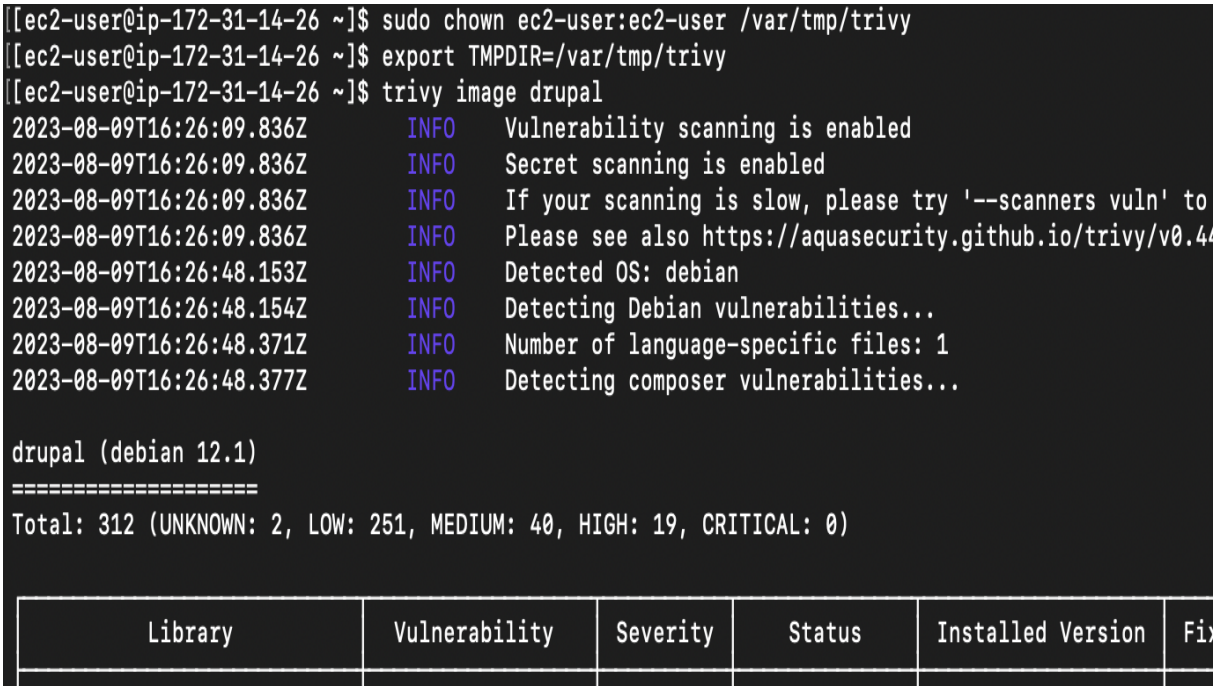
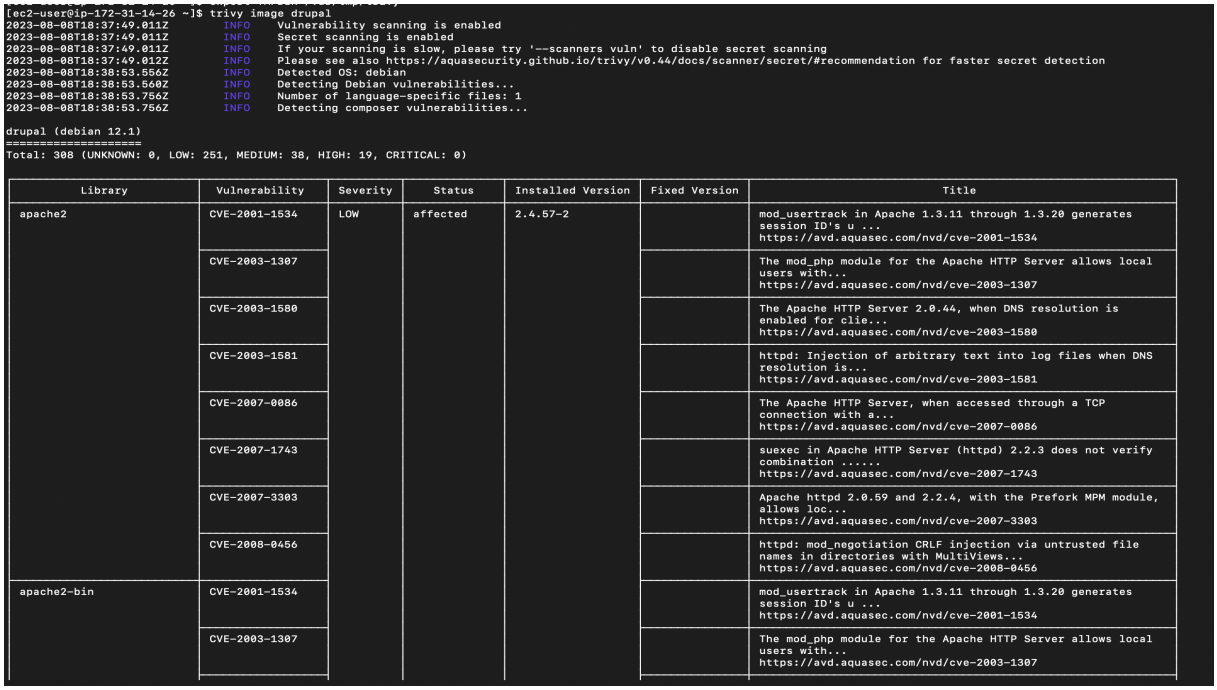


Figure 6.2: Vulnerabilities Identified before securing Drupal container.

6.2 Case Study/ Experiment-2

Next, is this important second experiment for research as it provides significant results in enhancing the security of container. Figure shows the activation of Apparmor profile using parser command line on Drupal container running on ubuntu EC2 instance. It can also be seen in the Figure that when user tries to access into the Drupal configured with Chef and apparmor is not allowed to run commands as the permission is denied by restricting polices and hence improving security of the container. Because when the container is configured alone with Chef it did not show any improvement but in collaboration of Apparmor, results came out fruitful. Also, Synk was configured to provide continuous monitoring of the container as can be seen in the Figure. Moreover, SElinux is enforced and its mode changed from permissive to enforced to enhance security as shown in Figure. Thus, it proves that this evaluation is successful as Apparmor, Chef, Synk, and Selinux secures container with customized configuration as per the requirements needed. So, the reduced vulnerabilities after securing the container can be seen in Figure vulner.



```
[ec2-user@ip-172-31-14-26 ~]$ trivy image drupal
2023-08-08T18:37:49.011Z        INFO    Vulnerability scanning is enabled
2023-08-08T18:37:49.011Z        INFO    Secret scanning is enabled
2023-08-08T18:37:49.011Z        INFO    If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2023-08-08T18:37:49.012Z        INFO    Please see also https://aquasecurity.github.io/trivy/v0.44/docs/scanner/secret/#recommendation for faster secret detection
2023-08-08T18:38:53.556Z        INFO    Detected OS: debian
2023-08-08T18:38:53.560Z        INFO    Detecting Debian vulnerabilities...
2023-08-08T18:38:53.756Z        INFO    Number of language-specific files: 1
2023-08-08T18:38:53.756Z        INFO    Detecting composer vulnerabilities...

drupal (debian 12.1)
====================
Total: 308 (UNKNOWN: 0, LOW: 251, MEDIUM: 38, HIGH: 19, CRITICAL: 0)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
apache2	CVE-2001-1534	LOW	affected	2.4.57-2		mod_usertrack in Apache 1.3.11 through 1.3.20 generates session ID's u ... https://avd.aquasec.com/nvd/cve-2001-1534
	CVE-2003-1307					The mod_php module for the Apache HTTP Server allows local users with... https://avd.aquasec.com/nvd/cve-2003-1307
	CVE-2003-1580					The Apache HTTP Server 2.0.44, when DNS resolution is enabled for clie... https://avd.aquasec.com/nvd/cve-2003-1580
	CVE-2003-1581					httpd: Injection of arbitrary text into log files when DNS resolution is... https://avd.aquasec.com/nvd/cve-2003-1581
	CVE-2007-0086					The Apache HTTP Server, when accessed through a TCP connection with a... https://avd.aquasec.com/nvd/cve-2007-0086
	CVE-2007-1743					suexec in Apache HTTP Server (httpd) 2.2.3 does not verify combination https://avd.aquasec.com/nvd/cve-2007-1743
	CVE-2007-3303					Apache httpd 2.0.59 and 2.2.4, with the Prefork MPM module, allows loc... https://avd.aquasec.com/nvd/cve-2007-3303
	CVE-2008-0456					httpd: mod_negotiation CRLF injection via untrusted file names in directories with MultiViews... https://avd.aquasec.com/nvd/cve-2008-0456
apache2-bin	CVE-2001-1534					mod_usertrack in Apache 1.3.11 through 1.3.20 generates session ID's u ... https://avd.aquasec.com/nvd/cve-2001-1534
	CVE-2003-1307					The mod_php module for the Apache HTTP Server allows local users with... https://avd.aquasec.com/nvd/cve-2003-1307

Figure 6.3: Vulnerabilities Reduced after securing Drupal Container.

Thus, these experiments showed critical results with defining how efficient the security tools are in securing the Drupal Docker containers and images that was the objective of the research question. So, the experiments show the importance of these tools used for vulnerability scanning and hardening of Drupal containers where Aqua Trivy and Anchore Engine were crucial for first experiment and Selinx, Apparmor, and Chef were outstanding in mitigating container risksZuppelli et al. (2023).

6.3 Discussion

The evaluation and case study experiments in this research illustrates deep insights in Docker containers, particularly showing the vulnerabilities present in Drupal containers running on Aws and exploring the effectiveness of various tools in addressing and mitigating these defects in container security. It is found that even Docker containers itself are not immune to vulnerabilities. This creates need to regular assessment of containers' security. For this reason, tools like Aqua Trivy, Anchore Engine, Chef, and Apparmor proved their strengths. In addition to this, it is found that when single tool is used, it did not improve security while when they used in combination it magnified their contribution in improving security. Also, the EC2 CPU utilization can be seen in Figure below. This research primarily focused into two divisions: one is to identify vulnerabilities in containers and second is to improve container security using multiple tools, eventually the proposed framework proved to be useful in achieving these objectives. Moreover, as the research progressed, use of certain tools got limited due to licensing restrictions. It is understood that there are some constraints when accessing premium tools to improve security.

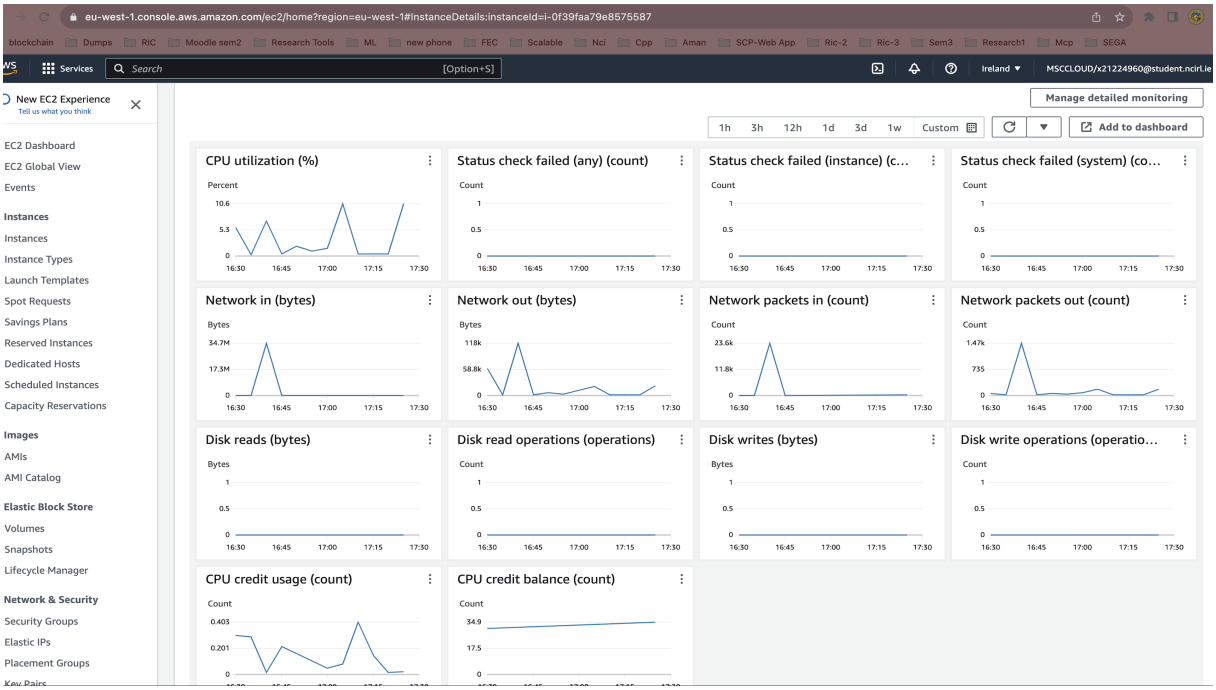


Figure 6.4: CPU utilization of Aws Linux instance.

However, even after hardening the system there is need for continuous monitoring of the system as containers face a constant threat and for that Synk was used in this research. Also, the research faced many challenges like licensing restrictions, if there was access to those tools that were mentioned in Design Specifications' section, results could have been different. However, the core strength of this research was its comparative approach. The approach was of first finding the vulnerabilities and then provides measures to rectify them. Therefore, this research has not only analysed vulnerabilities but also suggested practical solutions to mitigate them. Moreover, it should be noted that during implementation and testing, some lacks were identified in accuracy of the proposed model, it is not absolute, so it opens gates for further research in the domain of improving security of containers using multiple tools simultaneously.

Conclusion and Future Work

Security in Containers running on cloud is an on-going process as the technology keep advancing rapidly. As Docker is the most popular platform to use microservices like containers by organizations arises the need of crucial security of containers to avoid any data leaks. This research started from deep diving into identifying vulnerabilities in Drupal containers

running on Aws cloud to devising methodologies in order to mitigate them. It is understood that irrespective of the architecture and configuration of dockers, they are susceptible to vulnerabilities. Thus, a proactive approach to enhance security is imminent. Aqua Trivy, Anchore Engine, Chef, Synk, and Apparmor have proven their capabilities in enhancing container security and achieving the objectives defined in the research question. Moreover, it is learned that when there is a collaborative use of multiple tools the results are better than individual performance of security tools analyzed. Additionally, many security tools were tried but due to their limitation of use case, the research proceed with above ones. To summarize, this research exposes vulnerabilities in Drupal containers running on Aws cloud and evaluate multi-tool methodologies to mitigate them. Hence, presenting both a diagnostic and a prescriptive viewpoint for other researchers.

Future research can be done by extending this study by analyzing the licensed security tools that were attempted to use in this research. The opportunities for further enhancing and fostering the security could be done to have more secure Docker containers. Therefore, to conclude, it can be said that the research objectives that were defined in the research question have been fulfilled.

The link to the video presentation uploaded on youtube is here: `https://youtu.be/cxvgk0zNSCQ`

Acknowledgement

I am delighted to pay thanks to my Supervisor, **Mr Sean Heeney**, for his able guidance, feedback, effort and time. I am thankful for his valuable advice and continuous support throughout the journey of this research project.

Bibliography

Binnie, C. & McCune, R. (2021), *Container Image CVEs*, pp. 79–101.

Garg, S. & Garg, S. (2019), Automated cloud infrastructure, continuous integration and continuous delivery using docker with robust container security, *in* '2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)', pp. 467–470.

Gummaraju, J., Desikan, T. & Turner, Y. (2015), 'Over 30% of official images in docker hub contain high priority security vulnerabilities', *Technical Report* .

Javed, O. & Toor, S. (2021), An evaluation of container security vulnerability detection tools, *in* 'Proceedings of the 2021 5th International Conference on Cloud and Big Data Computing', pp. 95–101.

Merkel, D. et al. (2014), 'Docker: lightweight linux containers for consistent development and deployment', *Linux j* **239**(2), 2.

Patel, S. K., Rathod, V. R. & Parikh, S. (2011), Joomla, drupal and wordpress-a statistical comparison of open source cms, *in* '3rd International Conference on Trendz in Information Sciences & Computing (TISC2011)', IEEE, pp. 182–187.

Sengupta, R., Sai Prashanth, R. S., Pradhan, Y., Rajashekar, V. & Honnavalli, P. B. (2021), Metapod: Accessible hardening of docker containers for enhanced security, *in* '2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)', pp. 01–06.

Sultan, S., Ahmad, I. & Dimitriou, T. (2019), 'Container security: Issues, challenges, and the road ahead', *IEEE access* **7**, 52976–52996.

Wong, A. Y., Chekole, E. G., Ochoa, M. & Zhou, J. (2023), 'On the security of containers: Threat modeling, attack analysis, and mitigation strategies', *Computers & Security* **128**, 103140.

Zuppelli, M., Repetto, M., Caviglione, L. & Cambiaso, E. (2023), Information leakages of docker containers: Characterization and mitigation strategies, *in* '2023 IEEE 9th International Conference on Network Softwarization (NetSoft)', pp. 462–467.