Scan Docker images for vulnerabilities and implement security best practices

Table of Contents

Description

This guide explains how to secure Docker images and containers by scanning for vulnerabilities and applying security best practices. We'll use tools like **Docker Scan** and **Trivy** to detect vulnerabilities and suggest best practices for improving the security posture of Dockerized applications.

Problem Statement

Security vulnerabilities in Docker images can lead to serious risks in production environments. It's essential to scan images and follow best practices to minimize these risks. This guide will help you secure the **Java TodoApp** and **MySQL** containers by identifying vulnerabilities and hardening Docker security.

Prerequisites

Completion of all previous lab guides (up to Lab Guide-08) is required before proceeding with Lab Guide-09.

Software Requirement

- **Docker Desktop**: Installed on your Windows machine.
- **Trivy**: A vulnerability scanner for containers.
- **TodoAPP_MYSQl**: To download the source folder **click here**

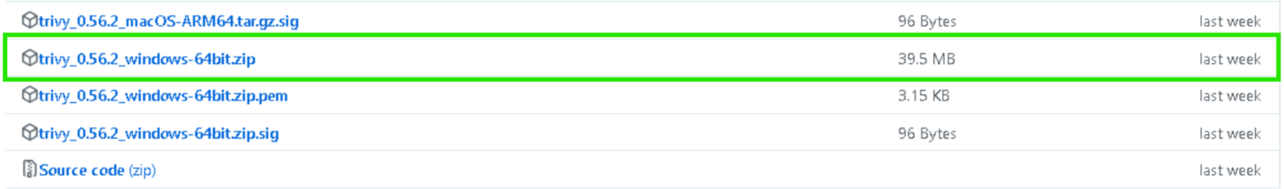
Hardware Requirement

- Minimum of 4 GB RAM
- At least 2 cores in the processor
- 5 GB of free storage space for Docker images and security tools

Trivy Installation Steps

1. Download the **Trivy binary for Windows**

- **Install Trivy** if you haven't already. You can download Trivy for Windows from the official GitHub repository.
- In the README file, click on download binary from https://github.com/aquasecurity/trivy/releases/latest/
- Download the appropriate Windows binary archive (**trivy_XXXX_Windows-64bit.zip**).



trivy_0.56.2_macOS-ARM64.tar.gz.sig	96 Bytes	last week
trivy_0.56.2_windows-64bit.zip	39.5 MB	last week
trivy_0.56.2_windows-64bit.zip.pem	3.15 KB	last week
trivy_0.56.2_windows-64bit.zip.sig	96 Bytes	last week
Source code (zip)		last week

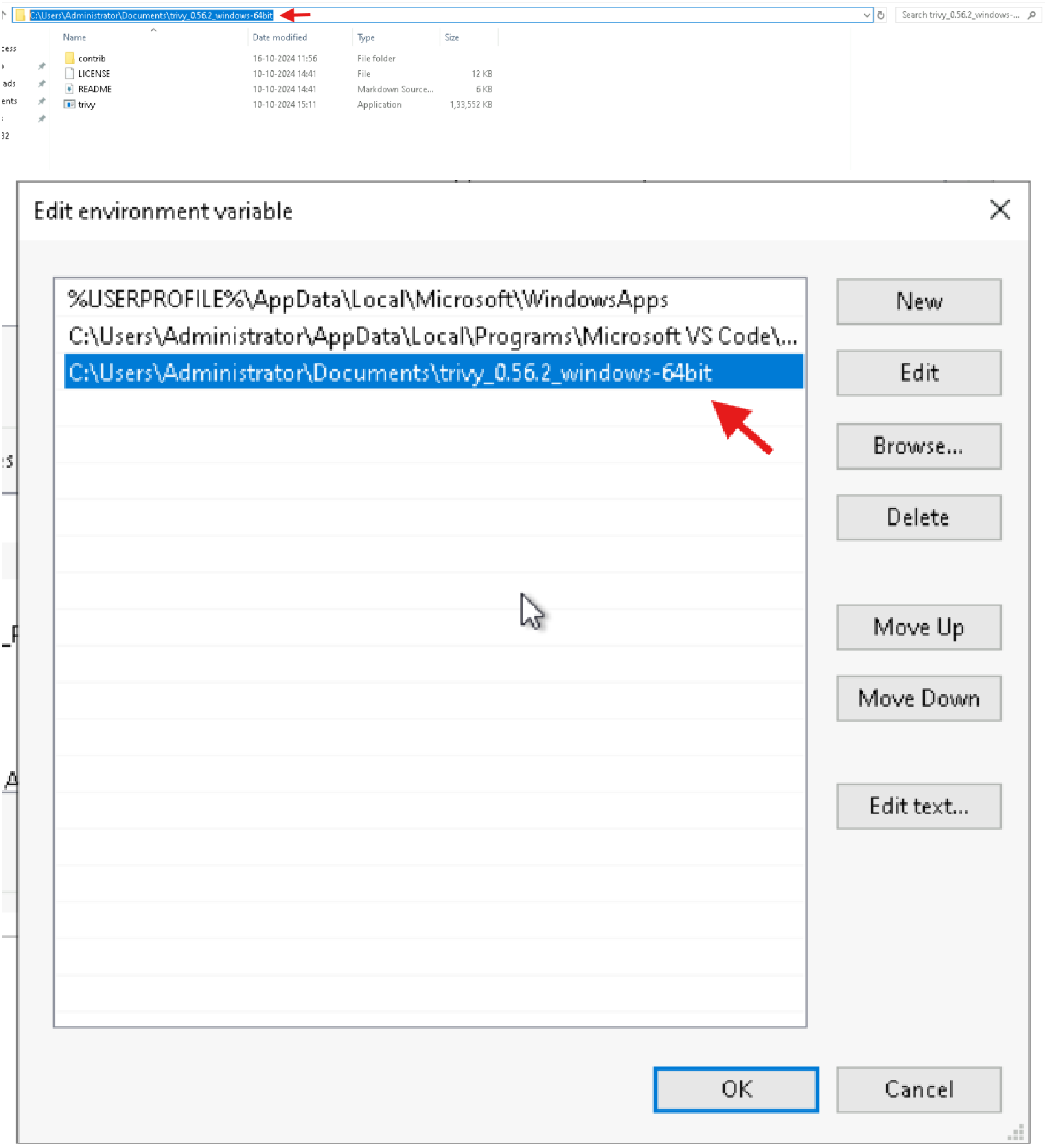
2. Unpack the archive

- Locate the downloaded **.zip** file and extract it using an archive tool like **WinRAR** or **7-Zip**.
- This will extract a folder containing the **trivy.exe** binary.

3. Add Trivy to your PATH

To use Trivy from any command prompt, add the folder where **trivy.exe** is located to your system's PATH:

- Right-click on **This PC** (or **My Computer**) and select **Properties**.
- Click on **Advanced system settings** on the left.
- In the **System Properties** window, click **Environment Variables**.
- Under **System variables**, find the **Path** variable, select it, and click **Edit**.
- Click **New** and paste the path where **trivy.exe** is located (e.g., **C:\path\to\trivy\folder**).
- Click **OK** to close all the windows.



C:\Users\Administrator\Documents\trivy_0.56.2_windows-64bit

Name	Date modified	Type	Size
contrib	16-10-2024 11:56	File folder	
LICENSE	10-10-2024 14:41	File	12 KB
README	10-10-2024 14:41	Markdown Source...	6 KB
trivy	10-10-2024 15:11	Application	1,33,552 KB

Edit environment variable

%USERPROFILE%\AppData\Local\Microsoft\WindowsApps

C:\Users\Administrator\AppData\Local\Programs\Microsoft VS Code\...

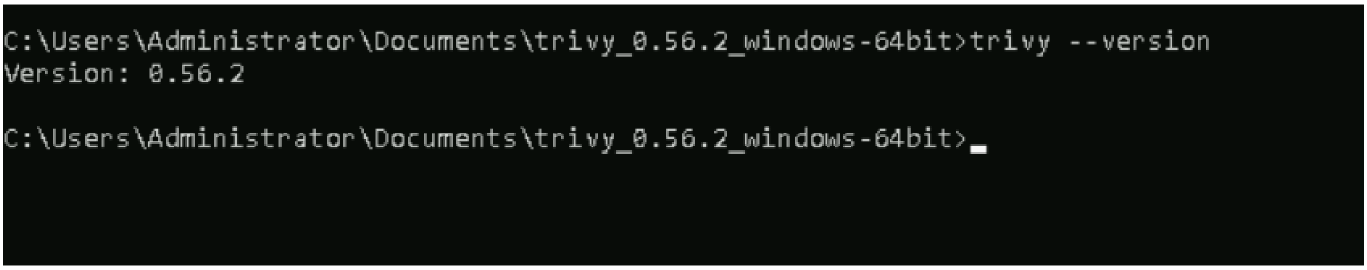
C:\Users\Administrator\Documents\trivy_0.56.2_windows-64bit

4. Verify the installation

Open **Command Prompt** and type the following to check if Trivy is installed correctly:

```
trivy --version
```

You should see the Trivy version printed, confirming the installation is successful.



```
C:\Users\Administrator\Documents\trivy_0.56.2_windows-64bit>trivy --version
Version: 0.56.2

C:\Users\Administrator\Documents\trivy_0.56.2_windows-64bit>
```

5. Set Execution Permissions (if necessary)

By default, Windows executables do not require explicit permission to run. However, ensure that the file is not blocked:

- Right-click on **trivy.exe**, go to **Properties**, and check if there is an "Unblock" button at the bottom of the window. If it exists, click **Unblock**, then **OK**.

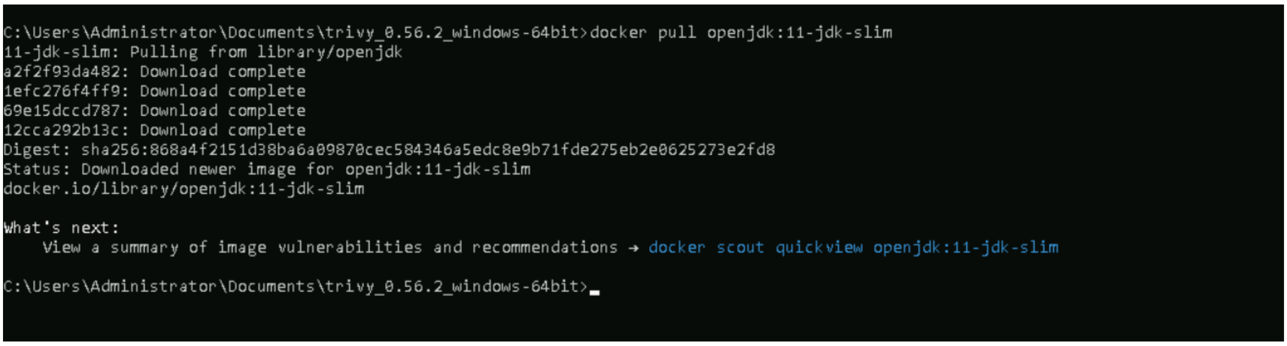
Implementation Steps

Step-1: Scan Docker Images for Vulnerabilities

1.1 Pull the Docker image

1. **Pull the image** you want to scan (e.g., MySQL or Java-based TodoApp):

   ```
   docker pull openjdk:11-jdk-slim
   docker pull mysql:8.0
   ```

   ```
   C:\Users\Administrator\Documents\trivy_0.56.2_windows-64bit>docker pull openjdk:11-jdk-slim
   11-jdk-slim: Pulling from library/openjdk
   a2f2f93da482: Download complete
   1efc276f4ff9: Download complete
   69e15dccd787: Download complete
   12cca292b13c: Download complete
   Digest: sha256:868a4f2151d38ba6a09870cec584346a5edc8e9b71fde275eb2e0625273e2fd8
   Status: Downloaded newer image for openjdk:11-jdk-slim
   docker.io/library/openjdk:11-jdk-slim

   What's next:
       View a summary of image vulnerabilities and recommendations → docker scout quickview openjdk:11-jdk-slim

   C:\Users\Administrator\Documents\trivy_0.56.2_windows-64bit>
   ```

2. **Scan the Docker image** using the built-in Docker Scan command:

   ```
   docker scan openjdk:11-jdk-slim
   ```

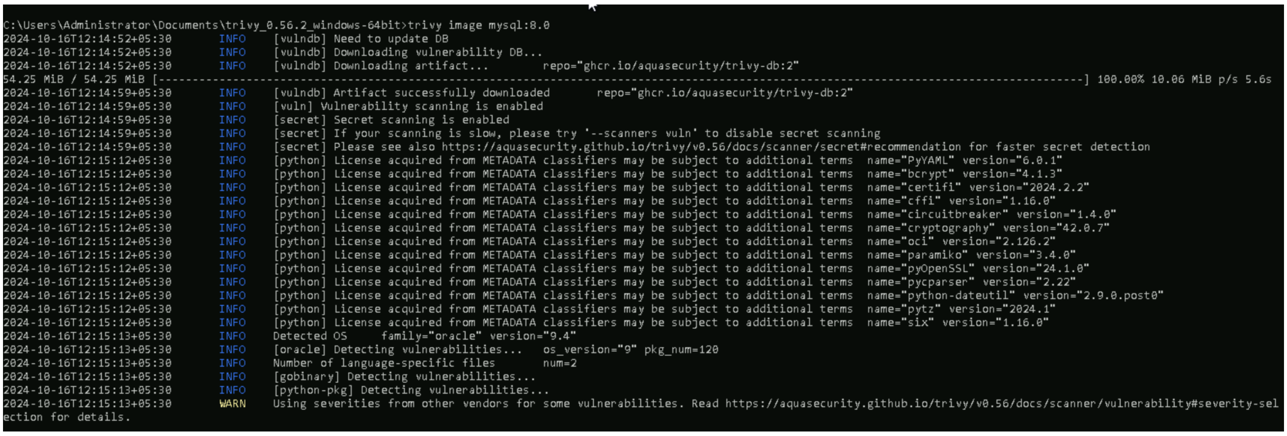
1.2 Using Trivy (Recommended)

1. **Install Trivy** if you haven't already. You can download Trivy for Windows from the official GitHub repository.

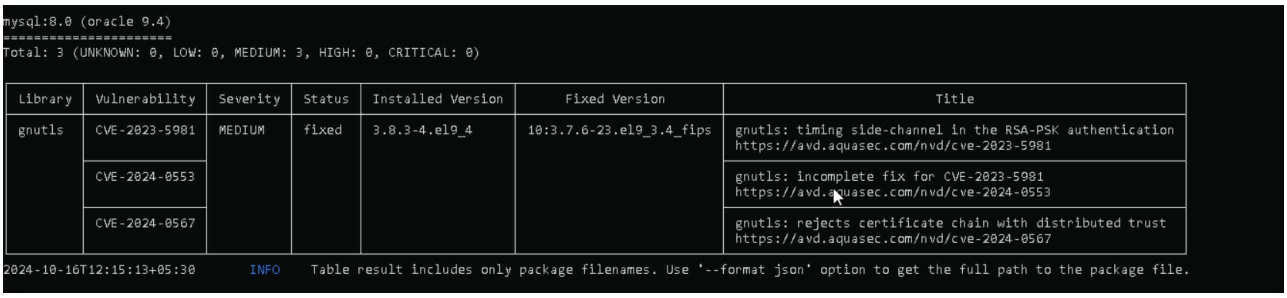
2. **Scan an image** for vulnerabilities:

```
trivy image mysql:8.0
```

Trivy will display a detailed report of vulnerabilities, categorized by severity (low, medium, high, critical).



```
C:\Users\Administrator\Documents\trivy_0.56.2_windows-64bit>trivy image mysql:8.0
2024-10-16T12:14:52+05:30       INFO    [vulndb] Need to update DB
2024-10-16T12:14:52+05:30       INFO    [vulndb] Downloading vulnerability DB...
2024-10-16T12:14:52+05:30       INFO    [vulndb] Downloading artifact...        repo="ghcr.io/aquasecurity/trivy-db:2"
54.25 MiB / 54.25 MiB [------------------------------------------------------------------------------------------------------------] 100.00% 10.06 MiB p/s 5.6s
2024-10-16T12:14:59+05:30       INFO    [vulndb] Artifact successfully downloaded       repo="ghcr.io/aquasecurity/trivy-db:2"
2024-10-16T12:14:59+05:30       INFO    [vuln] Vulnerability scanning is enabled
2024-10-16T12:14:59+05:30       INFO    [secret] Secret scanning is enabled
2024-10-16T12:14:59+05:30       INFO    [secret] If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2024-10-16T12:14:59+05:30       INFO    [secret] Please see also https://aquasecurity.github.io/trivy/v0.56/docs/scanner/secret#recommendation for faster secret detection
2024-10-16T12:15:12+05:30       INFO    [python] License acquired from METADATA classifiers may be subject to additional terms  name="PyYAML" version="6.0.1"
2024-10-16T12:15:12+05:30       INFO    [python] License acquired from METADATA classifiers may be subject to additional terms  name="bcrypt" version="4.1.3"
2024-10-16T12:15:12+05:30       INFO    [python] License acquired from METADATA classifiers may be subject to additional terms  name="certifi" version="2024.2.2"
2024-10-16T12:15:12+05:30       INFO    [python] License acquired from METADATA classifiers may be subject to additional terms  name="cffi" version="1.16.0"
2024-10-16T12:15:12+05:30       INFO    [python] License acquired from METADATA classifiers may be subject to additional terms  name="circuitbreaker" version="1.4.0"
2024-10-16T12:15:12+05:30       INFO    [python] License acquired from METADATA classifiers may be subject to additional terms  name="cryptography" version="42.0.7"
2024-10-16T12:15:12+05:30       INFO    [python] License acquired from METADATA classifiers may be subject to additional terms  name="oci" version="2.126.2"
2024-10-16T12:15:12+05:30       INFO    [python] License acquired from METADATA classifiers may be subject to additional terms  name="paramiko" version="3.4.0"
2024-10-16T12:15:12+05:30       INFO    [python] License acquired from METADATA classifiers may be subject to additional terms  name="pyOpenSSL" version="24.1.0"
2024-10-16T12:15:12+05:30       INFO    [python] License acquired from METADATA classifiers may be subject to additional terms  name="pycparser" version="2.22"
2024-10-16T12:15:12+05:30       INFO    [python] License acquired from METADATA classifiers may be subject to additional terms  name="python-dateutil" version="2.9.0.post0"
2024-10-16T12:15:12+05:30       INFO    [python] License acquired from METADATA classifiers may be subject to additional terms  name="pytz" version="2024.1"
2024-10-16T12:15:12+05:30       INFO    [python] License acquired from METADATA classifiers may be subject to additional terms  name="six" version="1.16.0"
2024-10-16T12:15:13+05:30       INFO    Detected OS     family="oracle" version="9.4"
2024-10-16T12:15:13+05:30       INFO    [oracle] Detecting vulnerabilities...   os_version="9" pkg_num=120
2024-10-16T12:15:13+05:30       INFO    Number of language-specific files       num=2
2024-10-16T12:15:13+05:30       INFO    [gobinary] Detecting vulnerabilities...
2024-10-16T12:15:13+05:30       INFO    [python-pkg] Detecting vulnerabilities...
2024-10-16T12:15:13+05:30       WARN    Using severities from other vendors for some vulnerabilities. Read https://aquasecurity.github.io/trivy/v0.56/docs/scanner/vulnerability#severity-sel
ection for details.
```



```
mysql:8.0 (oracle 9.4)
======================
Total: 3 (UNKNOWN: 0, LOW: 0, MEDIUM: 3, HIGH: 0, CRITICAL: 0)

┌─────────┬────────────────┬──────────┬────────┬───────────────────┬──────────────────────────┬─────────────────────────────────────────────────────────┐
│ Library │ Vulnerability  │ Severity │ Status │ Installed Version │      Fixed Version       │                          Title                          │
├─────────┼────────────────┼──────────┼────────┼───────────────────┼──────────────────────────┼─────────────────────────────────────────────────────────┤
│ gnutls  │ CVE-2023-5981  │ MEDIUM   │ fixed  │ 3.8.3-4.el9_4     │ 10:3.7.6-23.el9_3.4_fips │ gnutls: timing side-channel in the RSA-PSK authentication │
│         │                │          │        │                   │                          │ https://avd.aquasec.com/nvd/cve-2023-5981               │
│         ├────────────────┤          │        │                   │                          ├─────────────────────────────────────────────────────────┤
│         │ CVE-2024-0553  │          │        │                   │                          │ gnutls: incomplete fix for CVE-2023-5981                │
│         │                │          │        │                   │                          │ https://avd.aquasec.com/nvd/cve-2024-0553               │
│         ├────────────────┤          │        │                   │                          ├─────────────────────────────────────────────────────────┤
│         │ CVE-2024-0567  │          │        │                   │                          │ gnutls: rejects certificate chain with distributed trust │
│         │                │          │        │                   │                          │ https://avd.aquasec.com/nvd/cve-2024-0567               │
└─────────┴────────────────┴──────────┴────────┴───────────────────┴──────────────────────────┴─────────────────────────────────────────────────────────┘
2024-10-16T12:15:13+05:30       INFO    Table result includes only package filenames. Use '--format json' option to get the full path to the package file.
```

3. **Fix or mitigate vulnerabilities** based on the scan results. Update to the latest image versions or patch specific vulnerabilities as necessary.

1.3 Updating Vulnerable Images

If vulnerabilities are found, ensure you update the Docker images regularly. To pull the latest image versions:

```
docker pull mysql:latest
docker pull openjdk:latest
```

Step-2: Implement Docker Security Best Practices

2.1 Use Official and Verified Base Images

1. **Choose secure, official base images** from trusted sources, like the official **MySQL** or **OpenJDK** images:

```
FROM openjdk:11-jdk-slim
```

2. **Specify exact versions** for the base image to prevent using unstable or vulnerable versions:

```
FROM openjdk:11-jdk-slim@sha256:<specific_sha>
```

Example:

```
FROM openjdk:11-jdk-slim@sha256:
<868a4f2151d38ba6a09870cec584346a5edc8e9b71fde275eb2e0625273e2fd8>
```

2.2 Minimize Image Size

1. **Use smaller base images** such as **alpine** versions when possible to reduce the attack surface:

   ```
   FROM openjdk:11-jdk-alpine
   ```

   - **jdk-alpine**: Refers to an image based on **Alpine Linux**, a very lightweight Linux distribution.
   - **Alpine Linux** is a security-oriented, lightweight Linux distribution. It is often used in Docker images because it is:
     - **Minimal in size**: The Alpine-based image is significantly smaller (usually around 5 MB) compared to other distributions like Ubuntu or Debian. This reduces the size of your Docker image, which can lead to faster build and deployment times.
     - **Efficient**: Alpine uses the apk package manager, which is optimized for small and fast installations.
     - **Security-focused**: Alpine is designed to be more secure, with security features like stack-smashing protection and other hardening features built in.

2. **Remove unnecessary files and layers** in the Dockerfile:

   ```
   RUN apt-get update && apt-get install -y \
       some-package && \
       rm -rf /var/lib/apt/lists/*
   ```

   This removes unnecessary files after installation.

2.3 Run Containers as Non-Root Users

1. By default, Docker containers run as **root**, which is a security risk. To mitigate this, create and use a non-root user inside the container:

   ```
   RUN groupadd -r appgroup && useradd -r -g appgroup appuser
   USER appuser
   ```

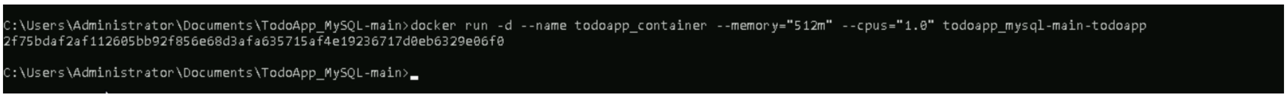
2. This will ensure the application runs under a less-privileged user, limiting the damage if an attacker gains access to the container.

2.4 Limit Resource Usage with Docker Flags

1. **Limit CPU and memory usage** for containers to prevent resource exhaustion attacks:

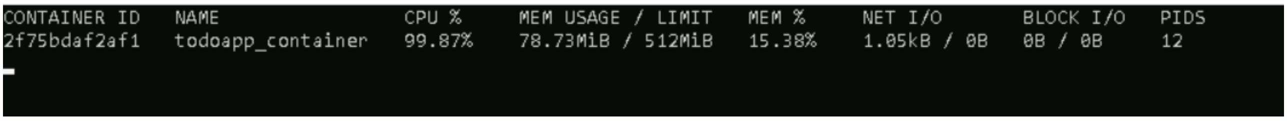
   ```
   docker run -d --name todoapp_container --memory="512m" --cpus="1.0" todoapp
   ```

   This command restricts the container to use no more than 512 MB of RAM and 1 CPU core.



```
C:\Users\Administrator\Documents\TodoApp_MySQL-main>docker run -d --name todoapp_container --memory="512m" --cpus="1.0" todoapp_mysql-main-todoapp
2f75bdaf2af112605bb92f856e68d3afa635715af4e19236717d0eb6329e06f0

C:\Users\Administrator\Documents\TodoApp_MySQL-main>
```



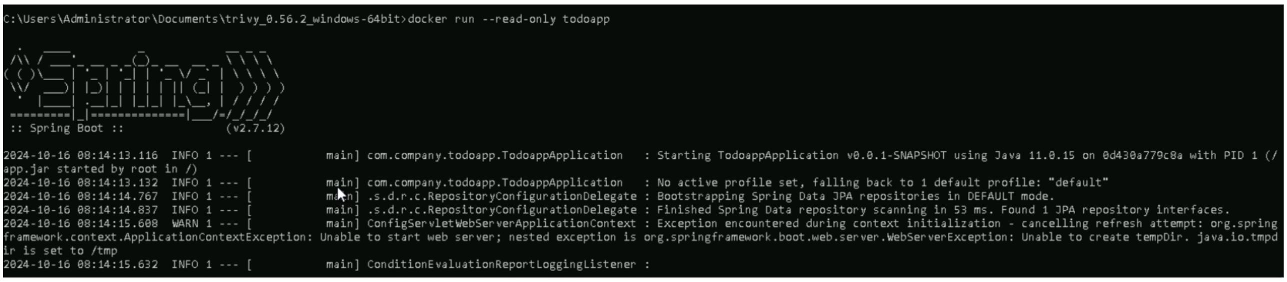
```
CONTAINER ID   NAME                CPU %     MEM USAGE / LIMIT     MEM %     NET I/O          BLOCK I/O   PIDS
2f75bdaf2af1   todoapp_container   99.87%    78.73MiB / 512MiB     15.38%    1.05kB / 0B      0B / 0B     12
```

2.5 Use Read-Only Filesystems

1. Make the container's filesystem read-only to prevent malicious code from modifying container files:

   ```
   docker run --read-only todoapp
   ```

   This prevents unwanted changes to the container's filesystem.



```
C:\Users\Administrator\Documents\trivy_0.56.2_windows-64bit>docker run --read-only todoapp

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::               (v2.7.12)

2024-10-16 08:14:13.116  INFO 1 --- [           main] com.company.todoapp.TodoappApplication   : Starting TodoappApplication v0.0.1-SNAPSHOT using Java 11.0.15 on 0d430a779c8a with PID 1 (/
app.jar started by root in /)
2024-10-16 08:14:13.132  INFO 1 --- [           main] com.company.todoapp.TodoappApplication   : No active profile set, falling back to 1 default profile: "default"
2024-10-16 08:14:14.767  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2024-10-16 08:14:14.837  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 53 ms. Found 1 JPA repository interfaces.
2024-10-16 08:14:15.608  WARN 1 --- [           main] ConfigServletWebServerApplicationContext : Exception encountered during context initialization - cancelling refresh attempt: org.spring
framework.context.ApplicationContextException: Unable to start web server; nested exception is org.springframework.boot.web.server.WebServerException: Unable to create tempDir. java.io.tmpd
ir is set to /tmp
2024-10-16 08:14:15.632  INFO 1 --- [           main] ConditionEvaluationReportLoggingListener :
```

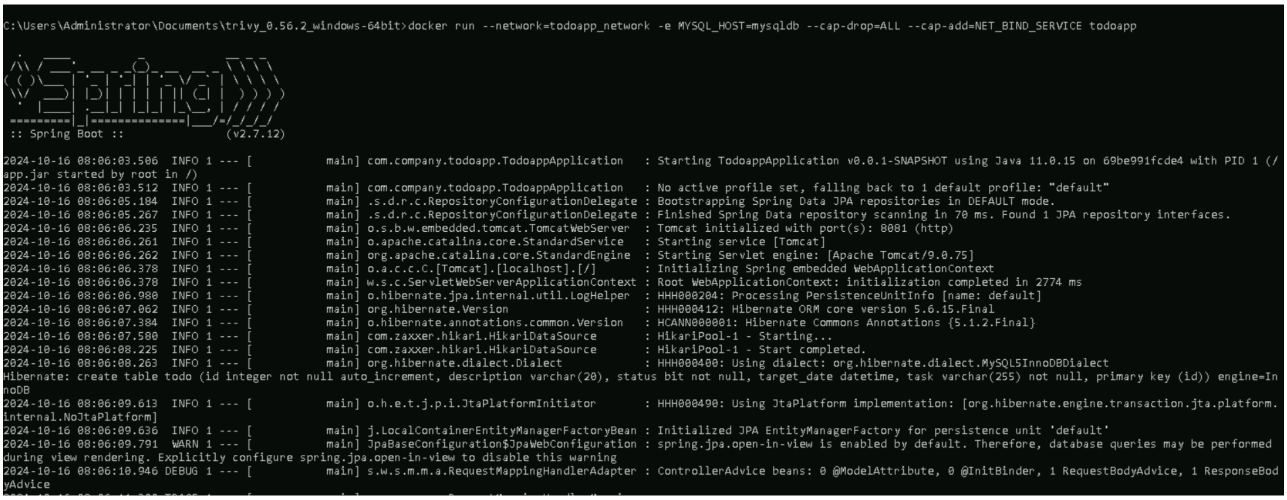
2.6 Limit Container Capabilities

1. **Drop unnecessary Linux capabilities** from the container, limiting its privileges:

   ```
   docker run --network=todoapp_network -e MYSQL_HOST=mysqldb --cap-drop=ALL --cap-add=NET_BIND_SERVICE todoapp
   ```

   This command drops all unnecessary capabilities, only allowing the container to bind to network ports.

   > Note: Make sure mysqldb is up and running in the same network as todoapp before running the above command



```
C:\Users\Administrator\Documents\trivy_0.56.2_windows-64bit>docker run --network=todoapp_network -e MYSQL_HOST=mysqldb --cap-drop=ALL --cap-add=NET_BIND_SERVICE todoapp

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::               (v2.7.12)

2024-10-16 08:06:03.506  INFO 1 --- [           main] com.company.todoapp.TodoappApplication   : Starting TodoappApplication v0.0.1-SNAPSHOT using Java 11.0.15 on 69be991fcde4 with PID 1 (/
app.jar started by root in /)
2024-10-16 08:06:03.512  INFO 1 --- [           main] com.company.todoapp.TodoappApplication   : No active profile set, falling back to 1 default profile: "default"
2024-10-16 08:06:05.184  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2024-10-16 08:06:05.267  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 70 ms. Found 1 JPA repository interfaces.
2024-10-16 08:06:06.235  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8081 (http)
2024-10-16 08:06:06.261  INFO 1 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-10-16 08:06:06.262  INFO 1 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.75]
2024-10-16 08:06:06.378  INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2024-10-16 08:06:06.378  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2774 ms
2024-10-16 08:06:06.980  INFO 1 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
2024-10-16 08:06:07.062  INFO 1 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 5.6.15.Final
2024-10-16 08:06:07.384  INFO 1 --- [           main] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
2024-10-16 08:06:07.580  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2024-10-16 08:06:08.225  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2024-10-16 08:06:08.263  INFO 1 --- [           main] org.hibernate.dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.MySQL5InnoDBDialect
Hibernate: create table todo (id integer not null auto_increment, description varchar(20), status bit not null, target_date datetime, task varchar(255) not null, primary key (id)) engine=In
noDB
2024-10-16 08:06:09.613  INFO 1 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.
internal.NoJtaPlatform]
2024-10-16 08:06:09.636  INFO 1 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2024-10-16 08:06:09.791  WARN 1 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed
during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2024-10-16 08:06:10.946 DEBUG 1 --- [           main] s.w.s.m.m.a.RequestMappingHandlerAdapter : ControllerAdvice beans: 0 @ModelAttribute, 0 @InitBinder, 1 RequestBodyAdvice, 1 ResponseBod
yAdvice
```

2.7 Network Security

1. **Isolate containers on a custom Docker network** and minimize exposure to external networks:

```yaml
networks:
  todoapp_network:
    driver: bridge
```

2. **Use Docker secrets** to store sensitive information like passwords, instead of environment variables.

References

- Docker Documentation: https://docs.docker.com/
- Trivy: https://github.com/aquasecurity/trivy
- Docker Security Best Practices: https://docs.docker.com/engine/security/