ISS - Docker Hardening - Coursework (cw4 - 40%)

1 Overall Context

Your team has been contracted by an organisation as Docker security specialists to harden a prototype web-application which is to run within Docker containers. This prototype is the first of several web-applications that the organisation will roll out over the next few months. The organisation is intending to run these under their own Docker runtime production environment.

The web-application development team have supplied you with the bespoke source files / scripts that should go into the build. They have also supplied associated Dockerfiles which should build containers that offer the necessary functionality. This is however, the first time the development team have been involved in a containerised deployment using Docker and they acknowledge that they have no experience of Docker security.

The production team plan to run the Docker Engine runtime (ie the production docker service) within a virtual machine which is dedicated to the task. This is likely to be a docker-ce 20.* version running on Centos 7. The production team have no experience of running Docker in production, however, they do have experience of deploying web-applications within conventional virtual machines offering platform as a service.

2 Your overall tasks

1. Determine your set of adequate test-case interactions. From these it should be possible to verify the extent to which each container, and the application as a whole, is both functional and secure.
2. Refine the supplied Dockerfiles, together with any custom files not available from official repositories, from which hardened images may be created which are able to run the application. The build process should be capable of running unattended so that it can be included into an automated build process. For the purposes of this assignment however, it is recognised that limited manual intervention may be required. The Dockerfiles, together with any additional scripts that you create, must be commented where you want to draw attention to the significant things that you are doing. They should also highlight the significant things you know should be done but you are not doing.
3. Generate hardened images from your refined Dockerfiles
4. Determine the runtime commands that are needed to run containers from your images, on the docker engine that the production team are planning to use. This should give the most secure implementation of the runtime functionality that you can achieve. These commands will occur at two distinct phases:
    - one-off configuration;
    - every time a container is run.
5. Verify that when you apply the run-time security that you are proposing to your containers, the application performs correctly against your set of interactions from Task 1.

3 Team roles

The tutor will give you a team-id. Within your team, each member should lead and be deputy for a fair division of the duties across tasks. It is expected that you will need to subdivide these primary tasks into sub tasks. Duties must be explicitly agreed and notified to the tutor prior to the end of term2 by email with all team members cc'd.

4 Deliverables

4.1 Succinct Report

Via tabula, a pdf report named `iss-cw4.pdf` containing five sections:

- section 1 tabulating Task 1 (set of adequate interactions).
- section 2 containing your reasoning and evidence for your submission against tasks 2 & 3 (image hardening & image generation).
- section 3 containing your reasoning and evidence for your submission against tasks 4 & 5 (runtime hardening & verification).
- section 4 covering anything else.
- References.

Use inline citation in comments in source files to identify the full bibliographic reference in the references section.

In sections 2, 3, & 4, you should illustrate significant aspects of the process that you followed. Include significant examples that demonstrate your ability to think and act coherently. It is not sufficient merely to produced the finished item; reasoning about the process followed is essential.

You should also identify the team member(s) who primarily did the work associated with each major chunk of work.

Apart from the coversheet, the reports of all team members should be identical

4.2 Compressed archive

Via Tabula, each team member will supply identical compressed archives named `iss-cw4.tar.gz` containing:

- the build directories and additional scripts related to the build process associated with Task2 1 & 2.the various files require to deploy the containers securely in the runtime environment associated with Task 3.

4.3 Verification hashes

Via Tabula, a file (named `iss-cw4-hashes.sha1`) that contains the sha1 hashes of the individual files contained within `iss-cw4.tar.gz`. This will be sampled at the demo to confirm no significant changes have been made between the submission and the demo / viva. One way to achieve this is via

```
find ./iss-cw4 -type f -print0 | xargs -0 sha1sum | tee iss-cw4-hashes.sha1
```

5 Assessment

The assessment will comprise two, highly interdependent parts:

- conventional submitted material.
- a short demo / viva where the team members collectively and individually must demonstrate their intellectual ownership of the submitted material.

5.1 Marks up to 50%

Marks up to 50% will be awarded for satisfying all of the following:

1. satisfying all the file name constraints and internal structure of all the deliverables.
2. partial reduction of the attack surfaces of the images, supported by clear narrative in the associated readme of the reasoning associated with trying to reduce the attack surface.
3. somewhat restrictive though overly permissive security policies supported by clear narrative in the associated pdf / comments of the reasoning associated with the development of the security policies.
4. somewhat restrictive though overly permissive runtime environment supported by clear narrative in the associated readme of the reasoning associated with trying to establish a secure runtime container ecosystem.
5. A functional though somewhat insecure application that can be run at the demo supported by clear narrative in the pdf / script comments of understanding of the shortcomings of the submission.

5.2 Marks up to 70%

Marks up to 70% will be awarded for satisfying all of the following:

6. satisfying all the file name constraints and internal structure requirements of all the deliverables.
7. substantial reduction of the attack surfaces of at least one image, supported by clear narrative in the associated readme of the reasoning associated with reducing the attack surface. Other images partially reduced, as described in the “up to 50%” band.
8. tight security policies with suitably restrictive runtime environment for at least one container supported by clear narrative in the associated readme of the reasoning associated with the development of the security policies and secure runtime container ecosystem. Other containers somewhat restricted as the “up to 50%” band.
9. Robust treatment of persistence,
10. A functional modestly secure application that can be run at the demo.

5.3 Marks up to 100%

Marks up to 100% will be awarded for satisfying all of the following:

11. satisfying all the file name constraints of all the deliverables.
12. complete reduction of the attack surfaces of the images, supported by robust evidence and clear narrative in the associated readme
13. tight security policies with suitably restrictive runtime environment of all containers supported by robust evidence and clear narrative in the associated readme.
14. Robust, clean, well structured, well-commented build scripts that could be safely passed to a third party to maintain and / or further develop.
15. An additional, equivalent hardened image / runtime using podman, together with an

evaluation of podman against docker in this context.

16. Deep insight and associated application to the problem at hand, that goes beyond the material taught in the module.
17. A functional, secure, robust application that can be run at the demo.

6 Important Constraints

a) All activity must be conducted legally and ethically.

b) This is an assignment that should be carried out strictly within your team. Do not negotiate with other teams how to carry this out.

c) All source material must be referenced using the Harvard referencing convention. Use comments to reference sources in config files.

d) In order to achieve a given mark, there must be consistency between the submission and evidence at the demo/viva. Evidence at the demo / viva is fundamentally of two types: firstly technical evidence via the execution of commands, observation of outputs etc; secondly intellectual ownership evidence through familiarity with all aspects of the submission.

e) Changes in hashes discovered at the viva will be penalised to a maximum equivalent of a 10 day late submission penalty.

f) The demo / vivas are **provisionally scheduled for 30th May - 2nd June 2023.** The detailed demo / viva schedule will be published on Moodle for the module.

g) The demo / vivas will take place via MSTeams. Each team member will therefore need to be able to run the containers on their personal computer and screen share via MSTeams.

h) **Failure to be present for the viva will result in a mark of zero**.

i) At the demo / viva, you must be fully familiar with all aspects of your submission that represent any change from the original starter pack. Evidence at the viva of lack of familiarity may be reported as possible academic misconduct. **Be sure you re-familiarise yourself with your submission shortly before your viva**.

j) The default assumption is that all members contribute equitably to the assignment. Where there is clear evidence at the viva that it would be grossly unfair to allocate the same mark to all members of the team, then individual marks will be allocated.

k) Your team id should be used in the name of any artefact that may need to exist in the same environment as submissions from other teams.

7 Submission Deadline

There will be two deadlines associated with this assignment. The mark for the assignment will be supplied within 20 days of the viva.

7.1 Initial file submission

Files associated with the main part of the submission are to be submitted to Tabula by **12:00 Thursday 25th May 2023**.

7.2 Viva schdule

This will be notified via Moodle nearer the time. It is **likely to fall** in the period **30th May - 2nd June 2023**.

8 Late Submission Penalties

Work that is received after the submission time (UK time), will be recorded as having arrived the next working day.

Work that is not submitted on Tabula by the deadline will be considered late. Penalties for lateness are applied at the rate of 5 percentage points per university working day after the due date, up to a maximum of 10 university working days late. After this period, the work will be counted as a non-submission.

9 Structure of submitted tar.gz archive

Your submission directories should be named and organised as follows. Items identified with *<italicised-label>* should have the content of < … > replaced with your specific text.

- iss-cw4.tar.gz
  - builds
    - *<image01>*
      - Dockerfile
      - *<files and directories to be copied into image01>*
      - docker_*<image01-service>*.te
      - docker_*<image01-service>*.json
      - *<other files related to imag01>*
    - *<image02>*
      - Dockerfile
      - *<files and directories to be copied into image02>*
      - docker_*<image02-service>*.te
      - docker_*<image02-service>*.json
      - *<other files related to imag02>*
    - …
  - scripts
    - build-script.sh
    - build-README
    - one-off-run-config-script.sh
    - repeated-run-script.sh
    - run-README
    - *<other scripts as needed - script name should clearly identify purpose>*

10 Submission Coversheet

Your submission coversheet on the pdf should have the following fields:

MODULE TITLE: Implementing Secure Systems

MODULE CODE: WM242-24 (cw4)

TEAM NAME: *<team-name>*

ID NUMBER: *<your-id>*

PARTNER ID NUMBER(S): *<p-id1>, <p-id2>*