HOCHSCHULE
RAVENSBURG-WEINGARTEN
UNIVERSITY
OF APPLIED SCIENCES

DockerSec:

Cybersecurity Training and Testing Environment

By Rémi JARDRET &
Stéphane SIMON

HOCHSCHULE
RAVENSBURG-WEINGARTEN
UNIVERSITY
OF APPLIED SCIENCES

01

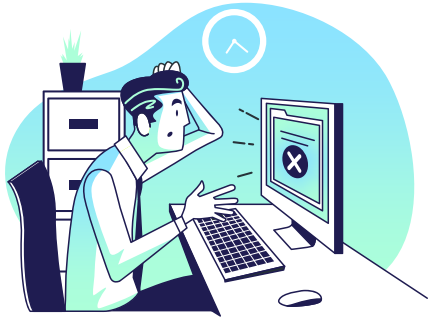
Introduction

Context, Motivation

Context

- Erasmus semester
- Project about system administration
- Already did in France with VM's architecture
- Wanted to make pentesting

Motivation



Problem

Project with VM :

- Heavy (20 GB/machine)
- Slow (6 VM running..)
- Long time to set up

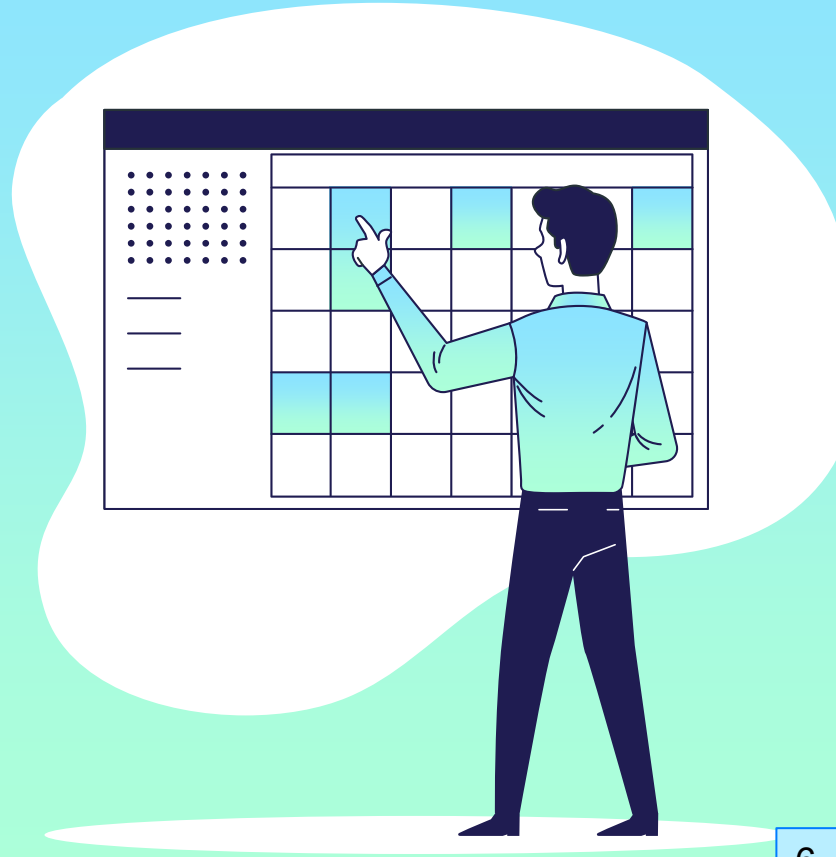
Solution

Project with Docker :

- Lighter
- Quick to install
- Easier to set up

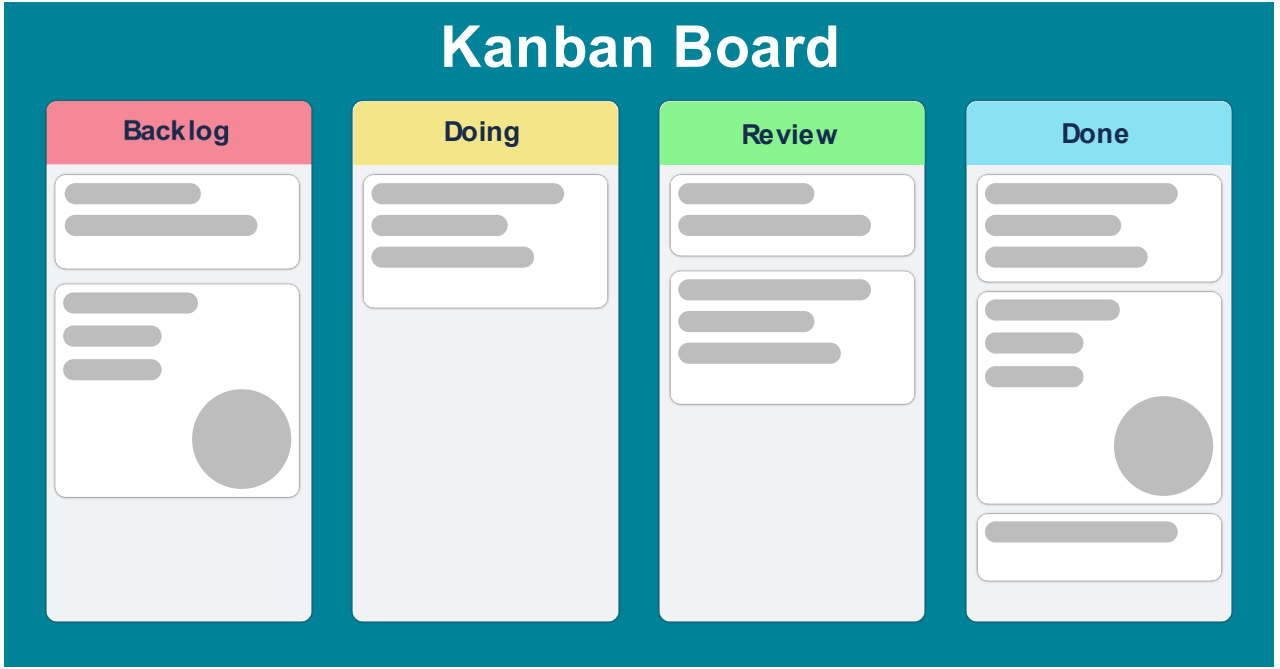
02 Project Management

Agile method, Kanban,
GANTT, Git



Kanban

- Agile methodology
- Visualization of progress
- Regular meetings
- Mixed with git tools and gantt method



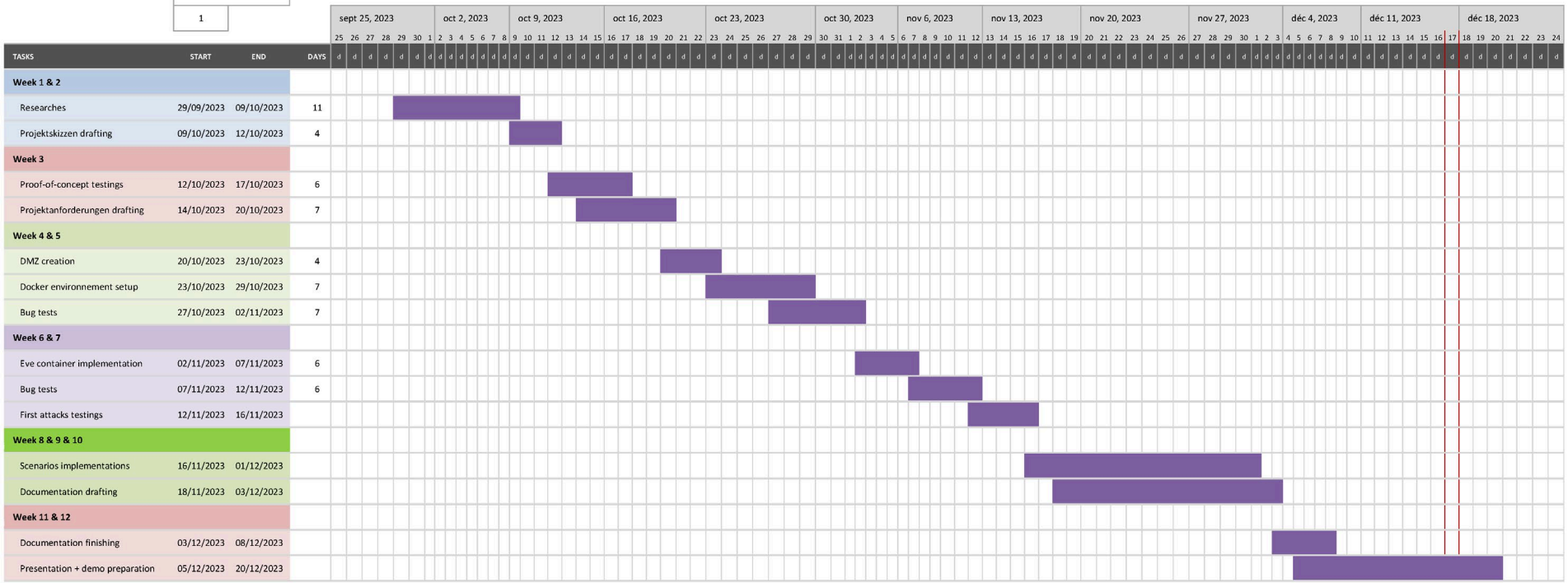
Gantt

DockerSec: Cybersecurity Training and Testing Environment

ESAIP & RWU

29/09/2023

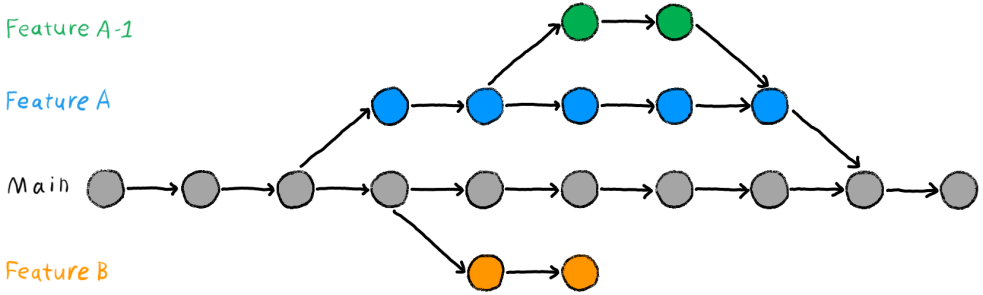
1



TASKS	START	END	DAYS
Week 1 & 2			
Researches	29/09/2023	09/10/2023	11
Projektskizzen drafting	09/10/2023	12/10/2023	4
Week 3			
Proof-of-concept testings	12/10/2023	17/10/2023	6
Projektanforderungen drafting	14/10/2023	20/10/2023	7
Week 4 & 5			
DMZ creation	20/10/2023	23/10/2023	4
Docker environnement setup	23/10/2023	29/10/2023	7
Bug tests	27/10/2023	02/11/2023	7
Week 6 & 7			
Eve container implementation	02/11/2023	07/11/2023	6
Bug tests	07/11/2023	12/11/2023	6
First attacks testings	12/11/2023	16/11/2023	
Week 8 & 9 & 10			
Scenarios implementations	16/11/2023	01/12/2023	
Documentation drafting	18/11/2023	03/12/2023	
Week 11 & 12			
Documentation finishing	03/12/2023	08/12/2023	
Presentation + demo preparation	05/12/2023	20/12/2023	

Git

- Collaborative project
- GitLab from University
- Branch per feature
- GitLab CI/CD to auto-compile our documentations

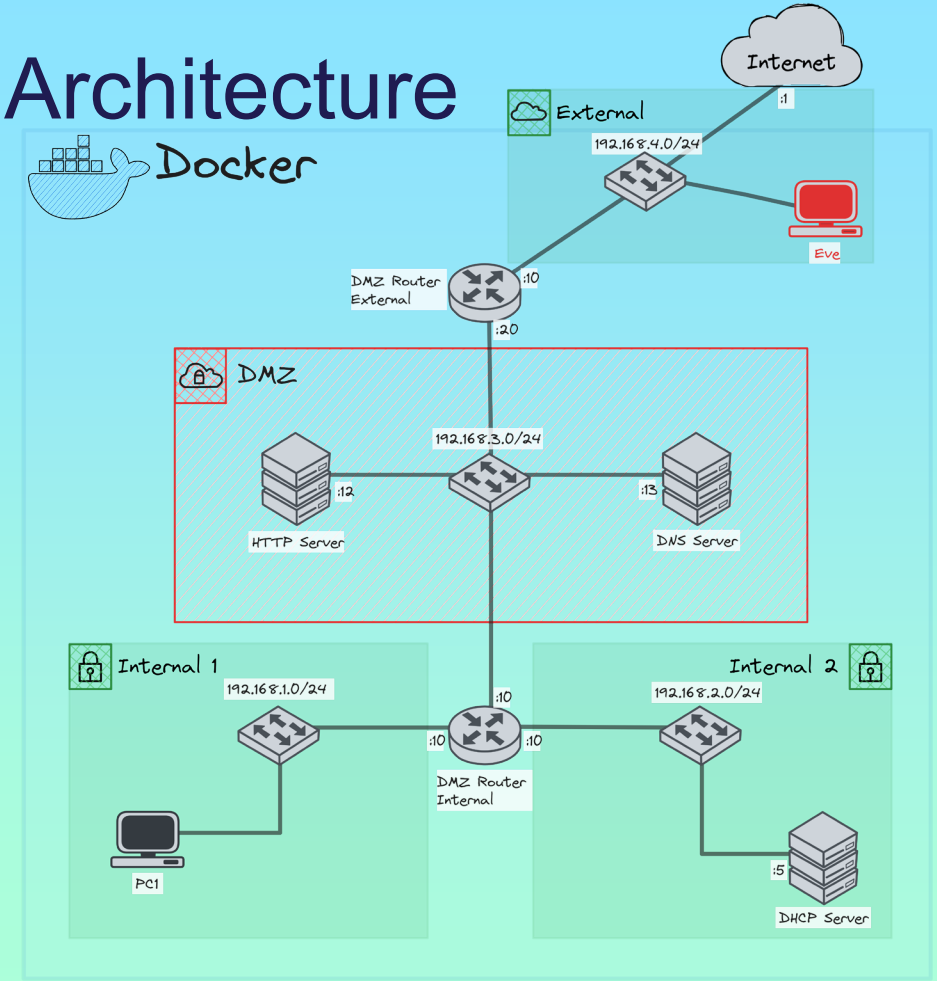


GitLab

03 Environment Architecture

Diagram of the project,
Structure

Network Architecture



Devices :
2x Routers
3x Servers
1x Client PC

Services :
DHCP
DNS
HTTP

Ubuntu

Every devices

Linux Administration

Metasploitable 2

Client PC

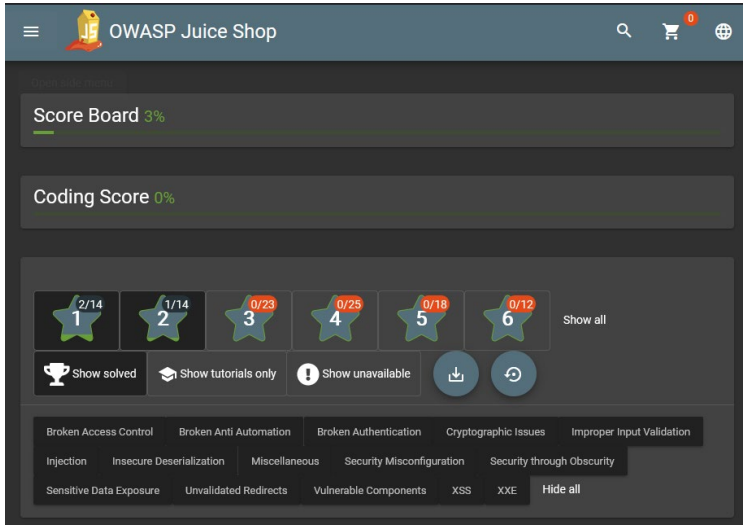
Vulnerabilities

Juice shop

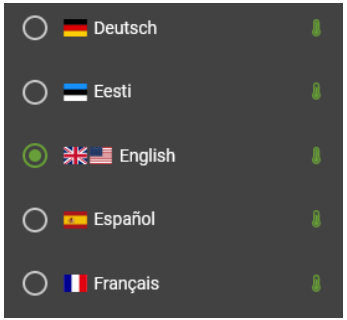
Server HTTP

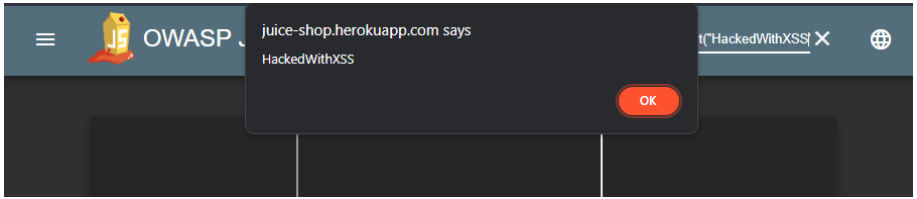
Web security

Juice-Shop


OWASP Juice Shop
Score Board 3%
Coding Score 0%
1 2/14
2 1/14
3 0/23
4 0/25
5 0/18
6 0/12
Show all
Show solved
Show tutorials only
Show unavailable
Broken Access Control
Broken Anti Automation
Broken Authentication
Cryptographic Issues
Improper Input Validation
Injection
Insecure Deserialization
Miscellaneous
Security Misconfiguration
Security through Obscurity
Sensitive Data Exposure
Unvalidated Redirects
Vulnerable Components
XSS
XXE
Hide all


OWASP Juice Shop
All Products
Apple Juice (1000ml)
1.99¤
Add to Basket
Apple Pomace
0.89¤
Add to Basket


Deutsch
Eesti
English
Español
Français

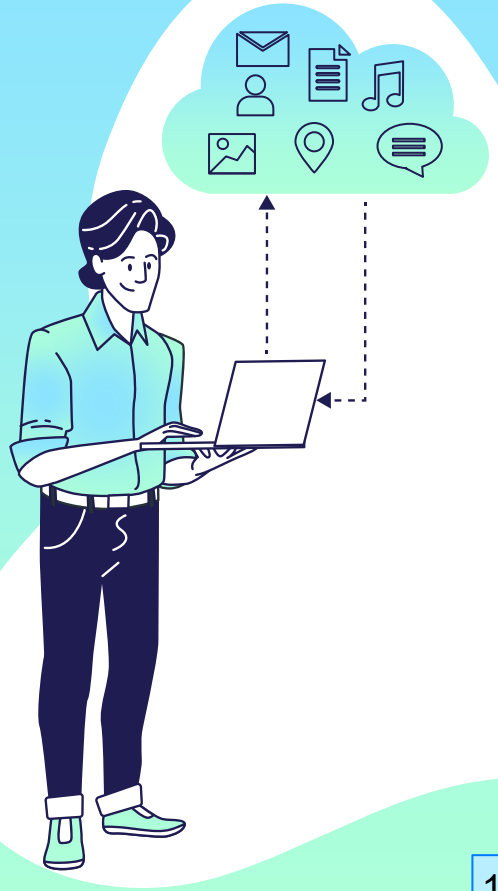

OWASP
juice-shop.herokuapp.com says
HackedWithXSS
OK


User Profile
Email:
admin@juice-sh.op
Username:
Set Username
File Upload:
Browse...
No file selected.
Upload Picture
or
Image URL:
Link Image

04

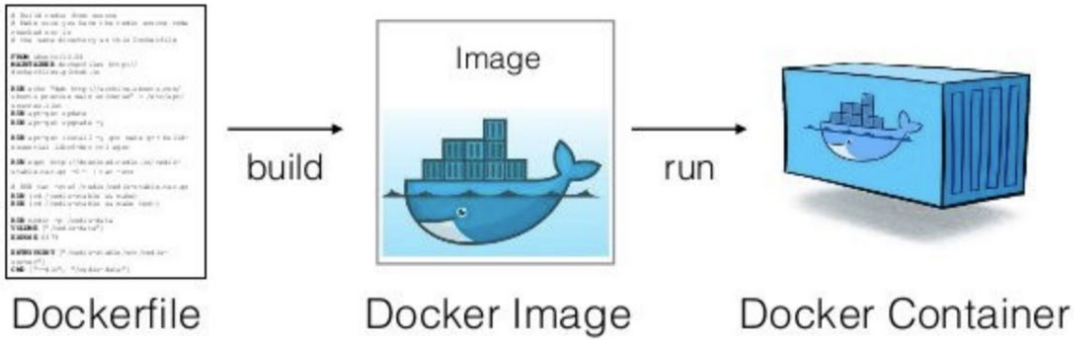
Implementation

Containers, DockerHub,
Docker Compose



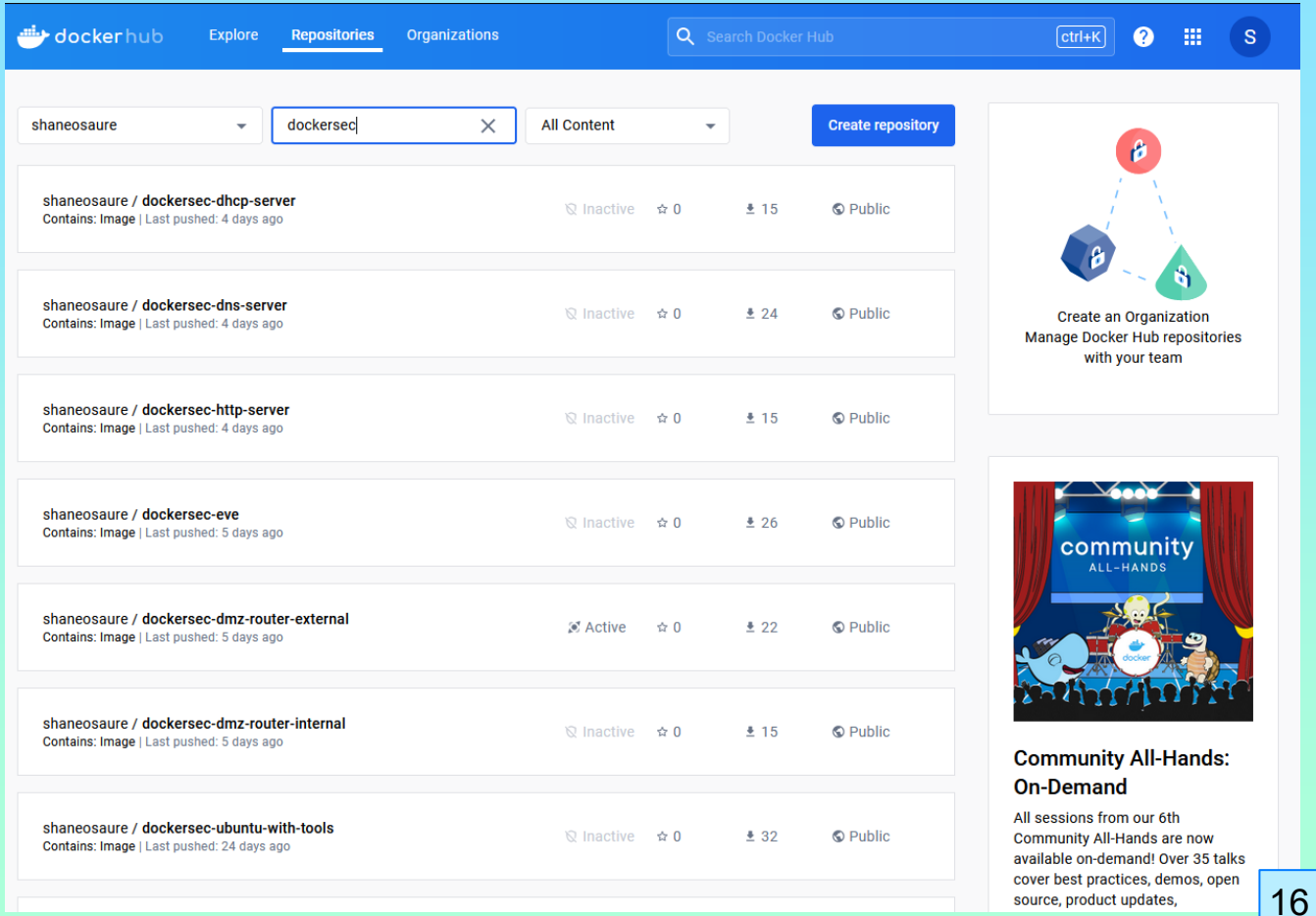
Docker's containers

- The tools we want
- The commands we want to do in the start of container
- The files we want to put in the container (like conf files)
- Easily runnable on every device / portability
- Easy reset

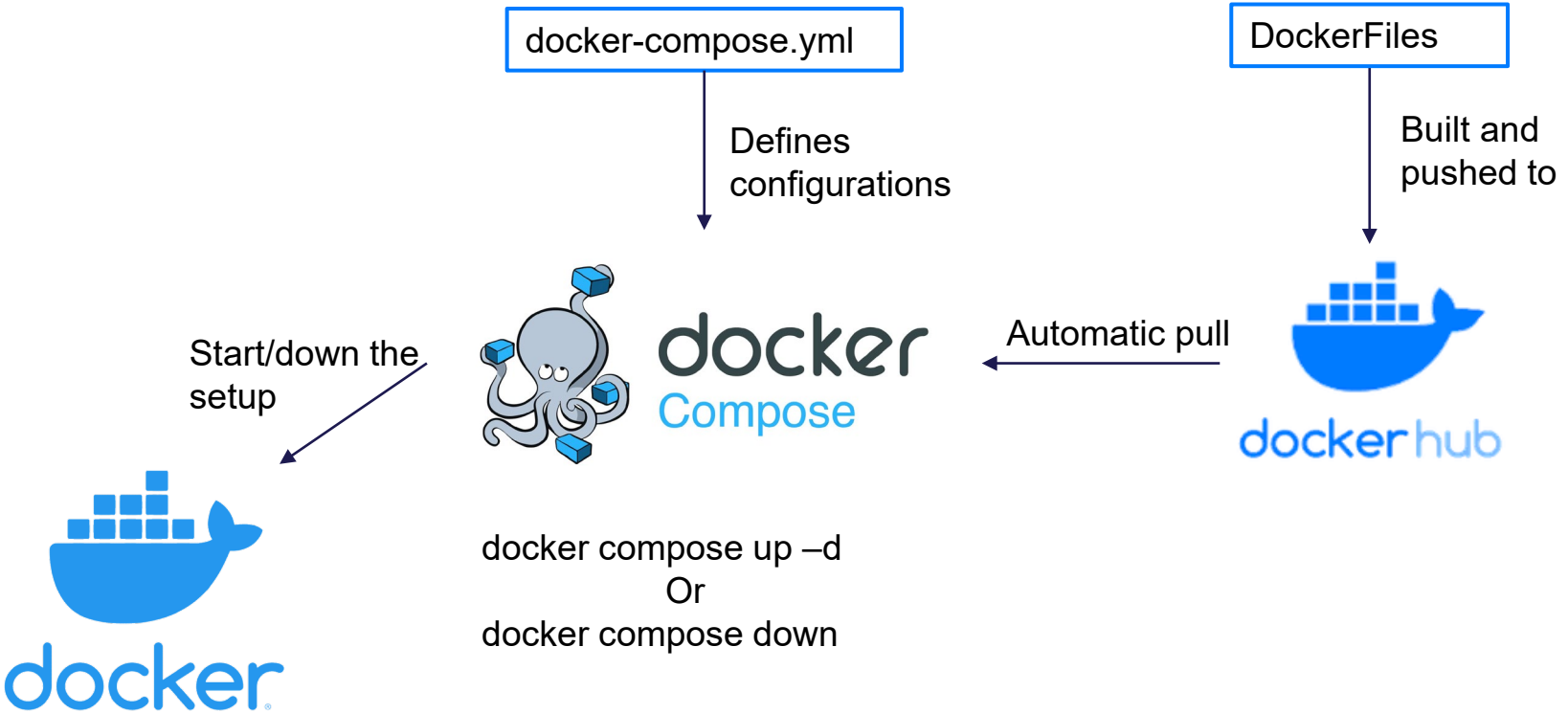


Dockerhub

- Where all the container's images are stored
- Everyone can get our project everywhere
- DockerScout functionality to keep project updated and secured
- Tags feature for multiple images



Docker Compose



05

Scenario

Documentations,
Demonstration

Documentations

Red Teaming Documentations

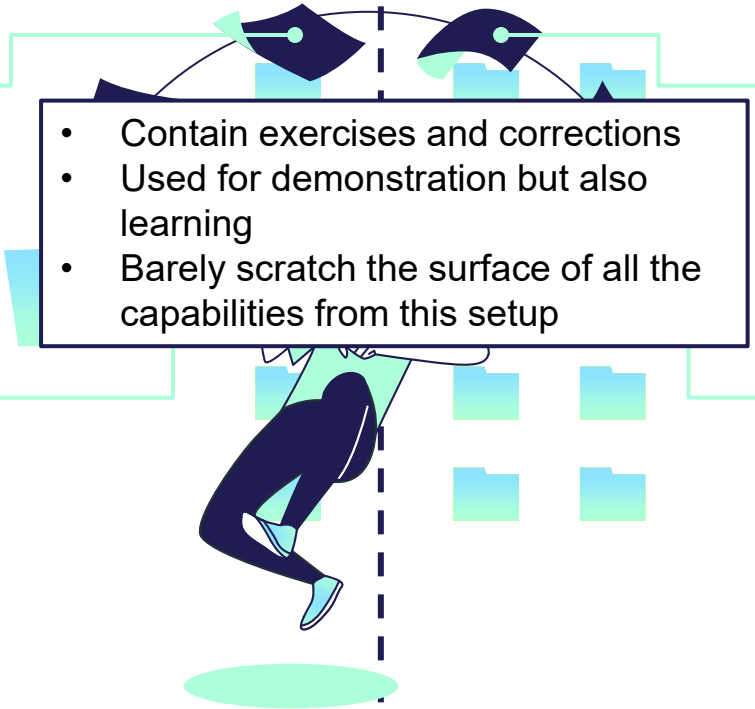
System Administration Documentations

Web security

With scenario 2

Pentesting

With scenario 1



- Contain exercises and corrections
- Used for demonstration but also learning
- Barely scratch the surface of all the capabilities from this setup

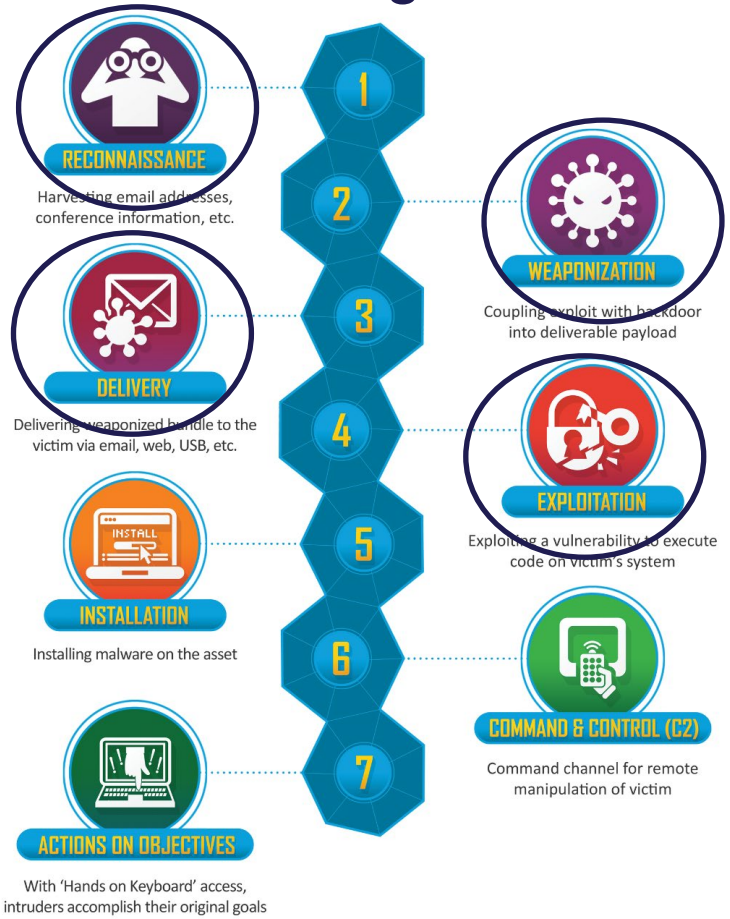
Firewalls & routing

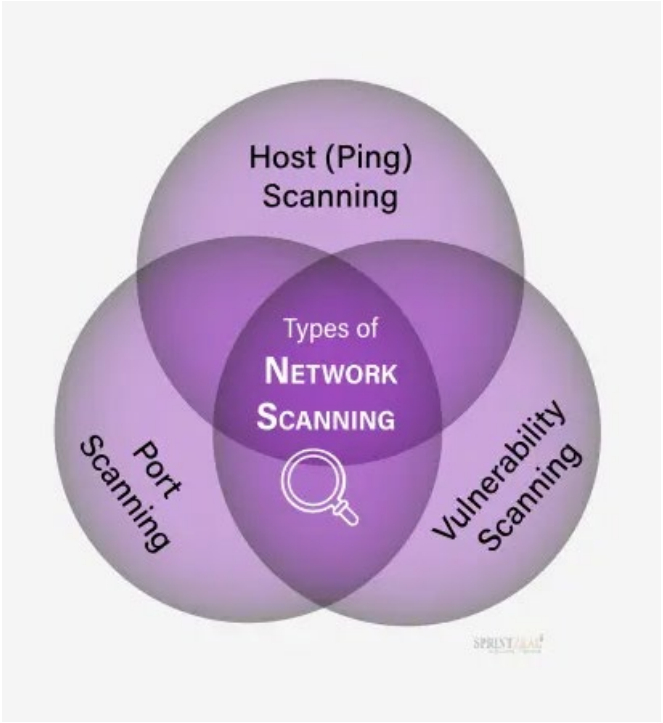
With part I

Configuring Services

With part II

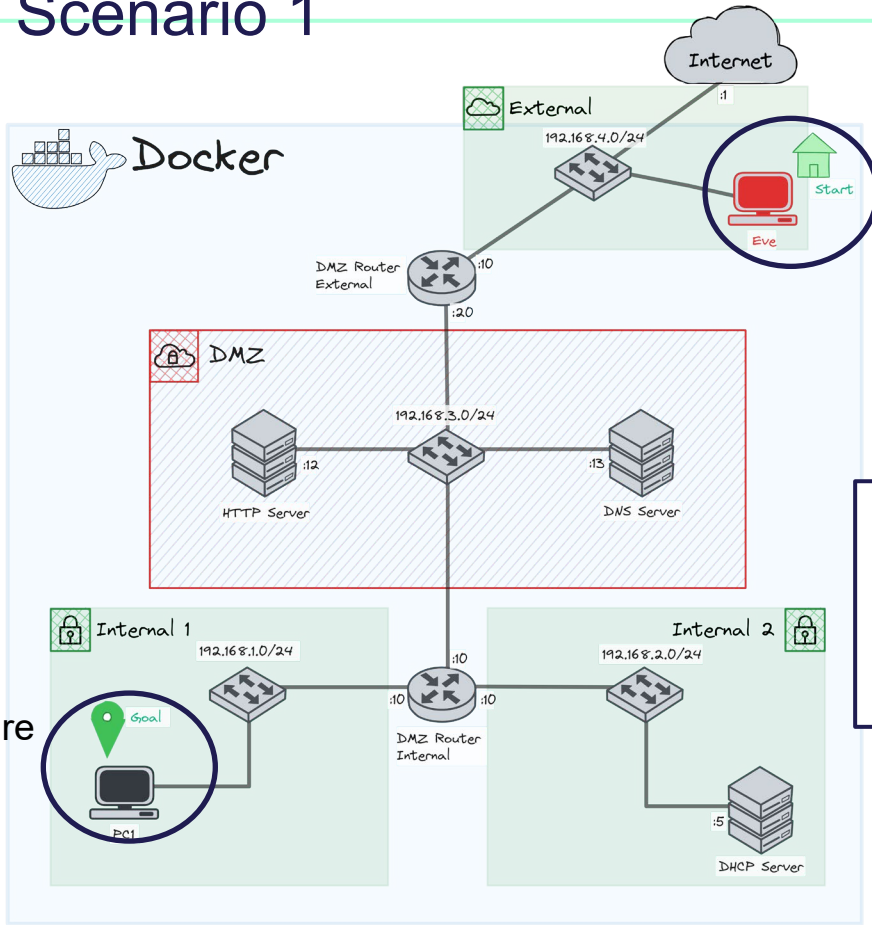
Red Teaming: Scenario 1





Red Teaming: Scenario 1

- Network Scanning
- Enumeration
- Vulnerabilities discovery
- Bruteforcing with dictionaries
- Payload for remote shell
- Privilege escalation

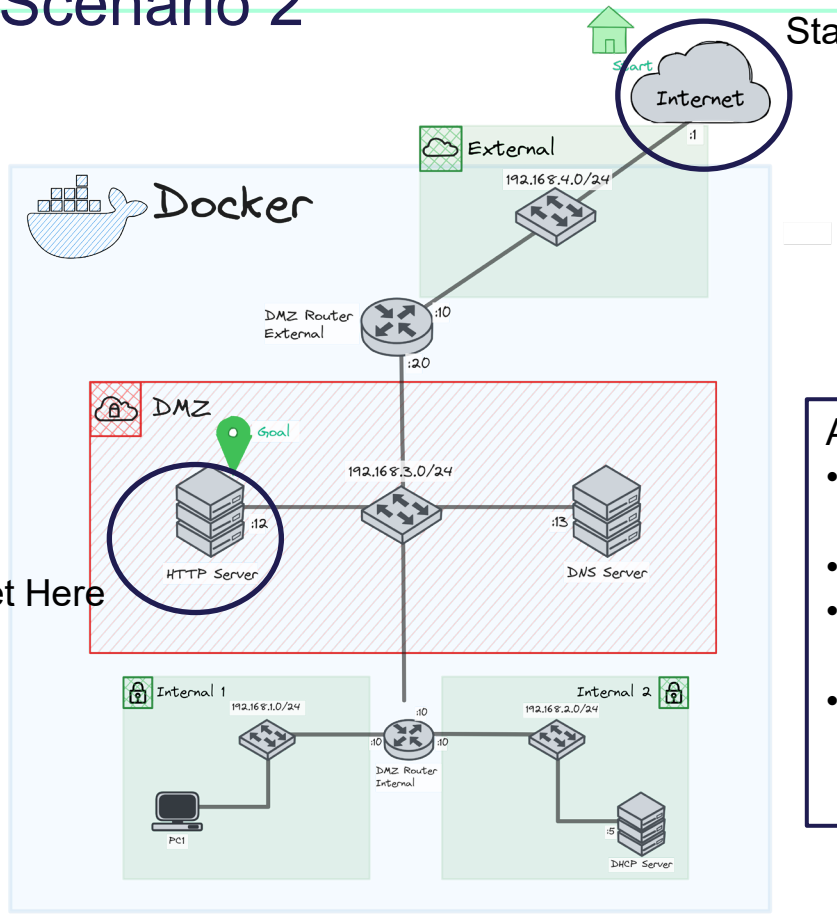


At the end:

- SSH stolen access
- Root rights
- Complete control of PC1

Red Teaming: Scenario 2

- DOM XSS Injections
- SQL Injection
- Poison Null Byte
- RCE with SSTi
- Reverse shell



At the end:

- You have access to a terminal
- Root rights
- Complete control of HTTP Server
- Can use the HTTP server to redo scenario 1

Red Teaming: exercises example

Reconnaissance

Scanning

Normally an attacker doesn't know the network infrastructure as you might be thanks to the schema of the system. Therefore, the attacker needs to do some network scanning in order to get more knowledge on the different hosts, services and firewall rules.

In this case, Eve is facing the first routeur "DMZ_External", therefore, it needs to find subnets attached to this routeur. To do so, we will use nmap, a powerful network scanning tool:

```
1 nmap <IP>/<MASK> -T5
```

Using this command, we will be able to list all the hosts that are on the given network, unless the firewall blocks the icmp-echo-request/replies.

Questions

1. How many hosts are up on the subnet 192.168.4.0/24 ?
2. Can you find any other subnets ?
3. Can you find PC1 IP address using nmap ?

Red Team Documentation 05.12.2023

Answers

1. If you run the nmap command on the subnet 192.168.4.0/24, you are supposed to have the following result:

```
└─# nmap 192.168.4.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-11 20:12 UTC
Nmap scan report for 192.168.4.1
Host is up (0.000059s latency).
All 1000 scanned ports on 192.168.4.1 are in ignored states.
Not shown: 991 filtered tcp ports (no-response), 9 filtered tcp ports (port-unreach)
MAC Address: 02:42:C5:37:9A:96 (Unknown)

Nmap scan report for source-DMZ_Router_External-1.source_External (192.168.4.10)
Host is up (0.000019s latency).
All 1000 scanned ports on source-DMZ_Router_External-1.source_External (192.168.4.10) are in ignored states
Not shown: 1000 closed tcp ports (reset)
MAC Address: 02:42:C0:A8:04:0A (Unknown)

Nmap scan report for 10a2dac2f0fe (192.168.4.3)
Host is up (0.0000090s latency).
All 1000 scanned ports on 10a2dac2f0fe (192.168.4.3) are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (3 hosts up) scanned in 6.07 seconds
```

As you can see, there are 2 hosts up (192.168.4.1 can be ignored, it is a docker bridge interface).

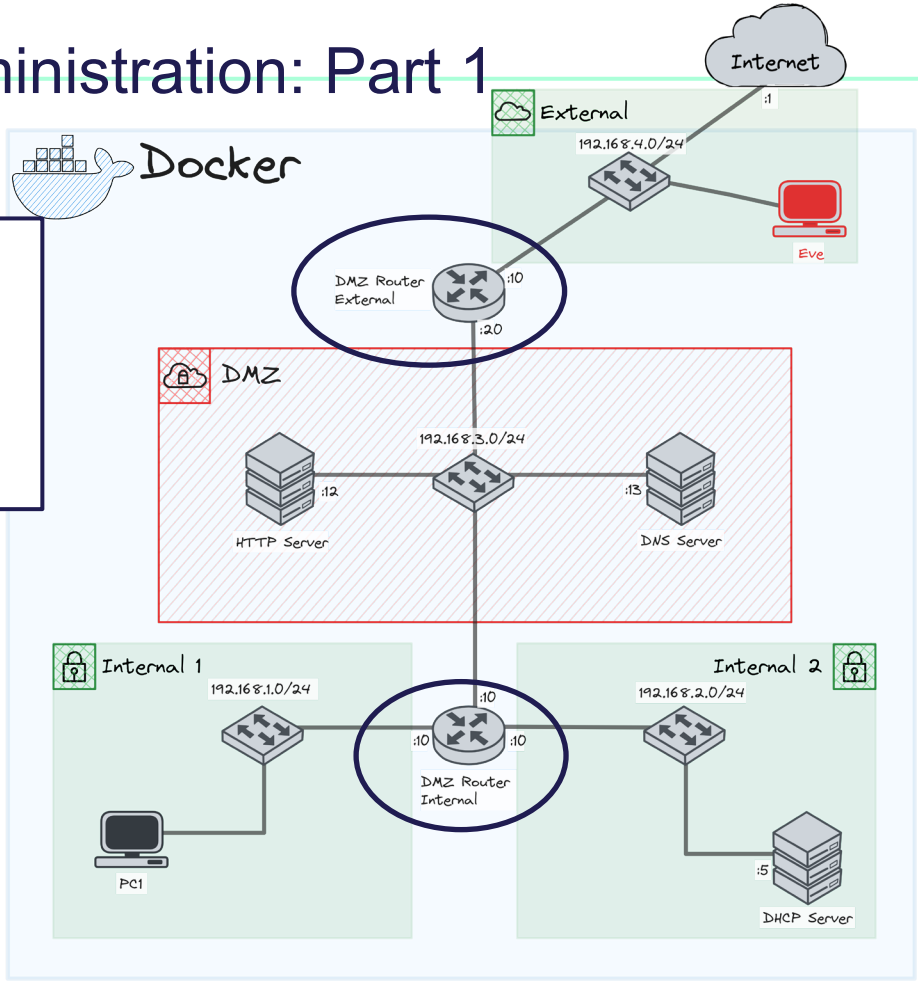
2. Using the nmap command you can specify other ip-ranges or masks. Therefore, using the subnet 192.168.1.0/21 for example, you can spot other Hosts. Don't forget, you are only observing the Hosts that **can** respond to pings. Some of them might be blocked be the firewalls, responding but never reaching Eve !

3. In order to find PC1, you need to use the option -Pn, this will allow you to omit the non-response of pings and still scan Hosts that might seem down. Using this command on 192.168.1.0/24 , you might have the following result:

```
└─# nmap 192.168.1.0/24 -T5 -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-11 20:14 UTC
Nmap scan report for 192.168.1.0
Host is up.
All 1000 scanned ports on 192.168.1.0 are in ignored states.
```

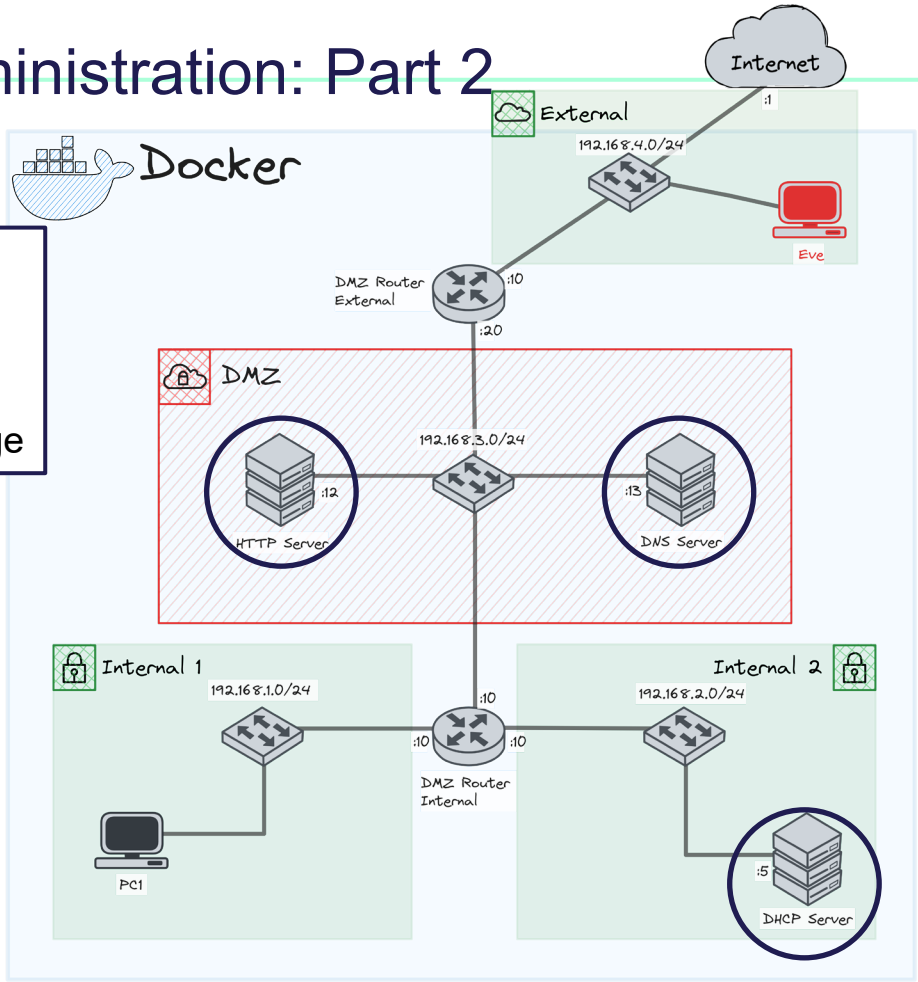
System Administration: Part 1

- Using IP tables
- Setting rules
- Stateful rules
- Using IP routes
- Gateways



System Administration: Part 2

- Checking processes
- Restarting services
- Explore logs
- Adding DNS zone
- Changing DHCP IP range



System Administration: exercises example

Here's the zone file of juiceshop:

```
$TTL 1D
@       IN      SOA     ns1.juiceshop.local. admin.juiceshop.local. (
                        2023111601 ; Serial
                        3H ; Refresh
                        15 ; Retry
                        1w ; Expire
                        3h ; Negative Cache TTL
);

@       IN      NS      ns1.juiceshop.local.
@       IN      A       192.168.3.12
www     IN      A       192.168.3.12
ns1     IN      A       192.168.3.13
```

Question

1. Create a zone for `example.com` and use `traceroute example.com` to show the proper configuration.

Answer

1. To create zone for example.com, you first need to create a zone file :

```
nano /etc/bind/zones/example.com
```

Then you will edit the file as follow:

```
$TTL 1D
@       IN      SOA     ns1.example.com. admin.example.com. (
                        2023111601 ; Serial
                        3H ; Refresh
                        15 ; Retry
                        1w ; Expire
                        3h ; Negative Cache TTL
);

@       IN      NS      ns1.example.com.
@       IN      A       192.168.4.3
www     IN      A       192.168.4.3
ns1     IN      A       192.168.4.3
```

Saving and exiting the file, you will now add the zone entry into `/etc/bind/named.conf.local`

```
zone "example.com" {
    type master;
    file "/etc/bind/zones/example.com";
};
```

Now you just have to restart the service using `service named restart` and check the output of `traceroute example.com`:

```
root@e528a68c75b3:/etc/bind# traceroute example.com
traceroute to example.com (192.168.4.3), 30 hops max, 60 byte packets
 1  192.168.3.20 (192.168.3.20)  0.716 ms  0.646 ms  0.596 ms
 2  192.168.4.3 (192.168.4.3)  0.562 ms  0.491 ms  0.436 ms
```

The IP is the one we have set into the DNS configuration, therefore it is now working !

Quick Demonstration

06

Conclusion

Final thoughts, future of project

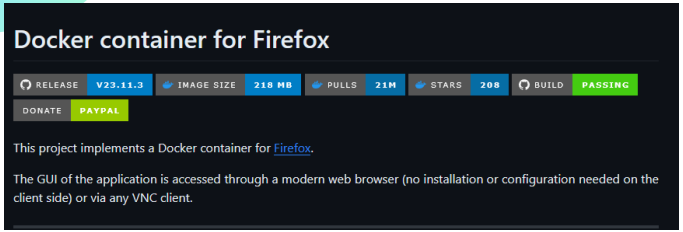
Final Thoughts

- Outperform standard VM setup
- Learnt a lot on docker
- Project management allowed continuous progression
- Complete documentations

- No Graphical applications
- More scenarios

Docker

Core of the project



Future of the project: community

OWASP Juiceshop

OWASP Juice Shop is probably the most modern and sophisticated insecure web application! It can be used in security trainings, awareness demos, CTFs and as a guinea pig for security tools! Juice Shop encompasses vulnerabilities from the entire OWASP Top Ten along with many other security flaws found in real-world applications!

through its community-led op
source software projects, hu
of chapters worldwide, tens
thousands of members, and
hosting local and global cont

Project Information

Flagship Project

Classification

Tool

Audience

Builder
Breaker
Defender

Road Map

jbarnett-r7 edited this page on Oct 12, 2016 · 2 revisions

This project is still in its infancy, but it should be initially useable for anyone looking to get their feet wet. Here are a few of the features that we would like to add in the near future (in no particular order):

- More vulnerabilities
  - This will be an ongoing feature that will only get better over time. We are constantly looking at new vulnerabilities to add to the base system. New scripts will be added to enable more vulnerabilities as time goes on, prioritized by how common they are.
- Expanding Operating Systems
  - First to tackle will be creating a Linux version of the system with its own set of vulnerabilities.
  - We also would like to expand to other Windows versions. Due to the nature of the scripts used to configure the system it should be fairly simple to port the existing vulnerabilities to a new OS.

Metasploitable

Docker Engine 24.0 release notes

This page describes the latest changes, additions, known issues, and fixes for Docker Engine version 24.0.

For more information about:

- Deprecated and removed features, see Deprecated Engine Features.
- Changes to the Engine API, see Engine API version history.

24.0.7

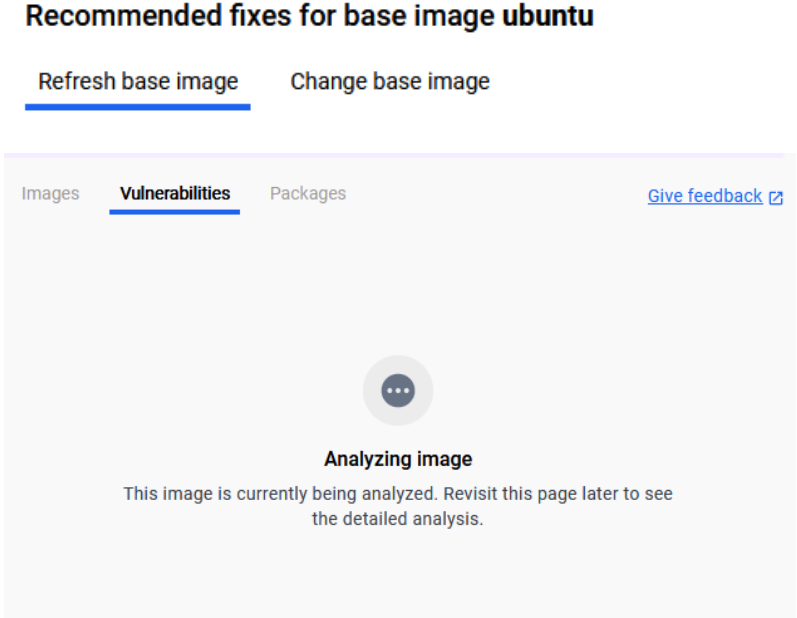
2023-10-27

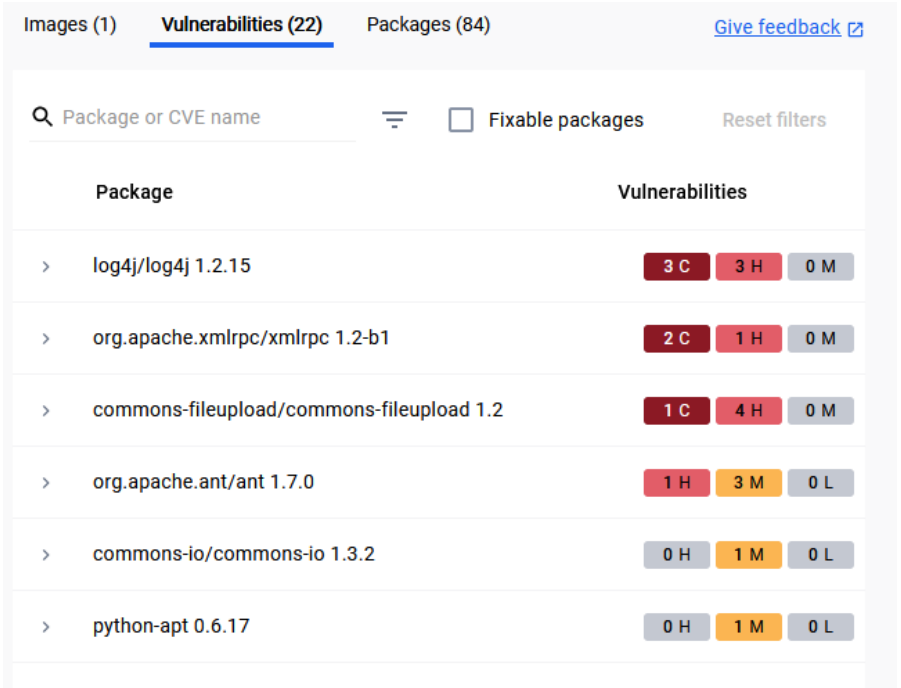
For a full list of pull requests and changes in this release, refer to the relevant GitHub milestones:

Docker

Future of the project: DockerScout

docker scout





HOCHSCHULE
RAVENSBURG-WEINGARTEN
UNIVERSITY
OF APPLIED SCIENCES

Thank you for your attention !