DOCKER HOST ATTACKS
Container/Host Security

Docker Host Attacks

A Docker host is a machine on which the Docker daemon and Docker containers run. Once the Docker host is compromised, the attacker can also access all the other containers running on it.
This section covers Docker socket misconfigurations that can be exploited by attackers to perform privilege escalation and take over the docker host. Scenarios, where insecurely configured host management tools can be leveraged to compromise the host, are also covered.

What will you learn?

- Exploiting misconfigurations to perform privilege escalation on Docker host
- Pwning a Docker host using insecure management tools
- Leveraging low-level components of the Docker ecosystem to take over the Docker host

References:

1. Docker Security CheatSheet (https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Docker_Security_Cheat_Sheet.md)

Labs:

- Misconfigured Docker Socket

  In this lab, you will learn to perform privilege escalation on a Docker host machine using exposed Docker TCP socket. A non-exhaustive list of activities to be covered includes:
  - Configure docker client to use TCP socket instead of UNIX socket
  - Use docker client to list Docker images present on Docker host
  - Run a container with host filesystem mounted to it
  - Use chroot to breakout of the container

- Weakest Link

  In this lab, you will learn to take over a Docker host by leveraging admin access to the Portainer web UI. A non-exhaustive list of activities to be covered includes:
  - Use Portainer web UI access to launch privileged container
  - Mount host disk inside the container
  - Use chroot to break out of the container
  - Run commands on host machine

- Weakest Link II

  In this lab, you will learn to take over a Docker host by leveraging weak admin credentials of Portainer. A non-exhaustive list of activities to be covered includes:
  - Perform dictionary attack to find password of Portainer admin user
  - Use Portainer web UI access to launch container with host filesystem mounted on it
  - Mount host disk inside the container
  - Use chroot to break out of the container
  - Run commands on host machine

- Leveraging Containerd

  In this lab, you will learn to use containerd to perform privilege escalation when the user is not allowed to use Docker. A non-exhaustive list of activities to be covered includes:
  - Start a container with host filesystem mounted (bind mount) to it using containerd
  - As an alternate approach, start a privileged container using containerd
  - Use DAC_READ_SEARCH to read the files from root filesystem

process memory to retrieve artifacts. A non-exhaustive list of activities to be covered includes:

- Start a privileged container using containerd
- Compile a reverse shell payload as kernel module
- Inject kernel module into host machine kernel
- Get reverse shell session to execute commands on host machine
- Dump process memory and retrieve the artifact

- Low-Level Container Runtime

  In this lab, you will learn to leverage runc to perform privilege escalation when the user is not able to use Docker or containerd. A non-exhaustive list of activities to be covered includes:
  - Create runc spec for container with host filesystem mounted on it
  - Run container using runc
  - Access files of host machine using container

Misconfigured Docker Socket

Start

Weakest Link

Start

Weakest Link II

Start

Leveraging Containerd

Start

Low-Level Container Runtime

Start

Leveraging Containerd II

Start

Abusing Group Membership

Start

Exploiting Remote Docker Host

Start