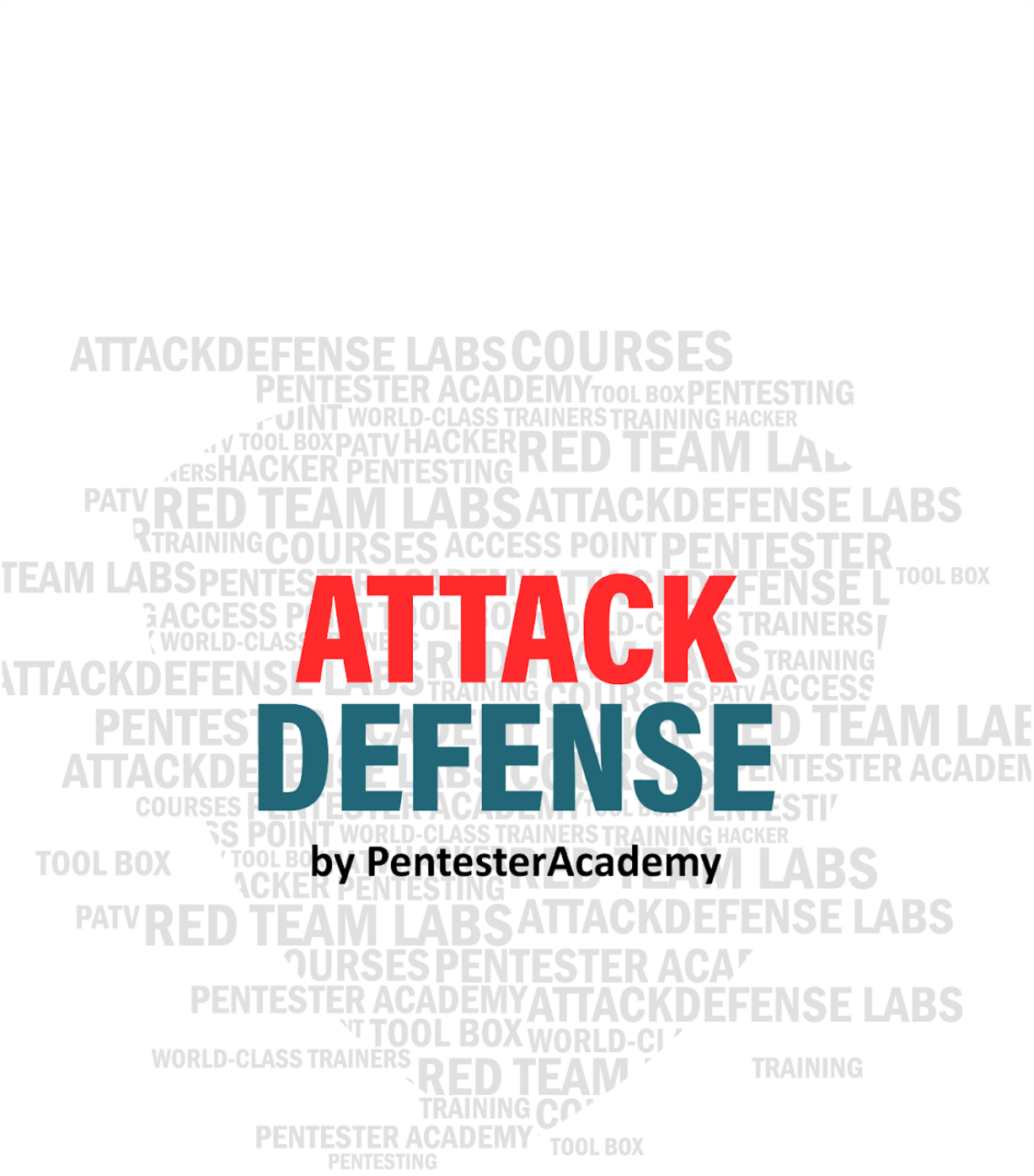
ATTACK
DEFENSE
by PentesterAcademy

Name	Docker Bench Security
URL	https://attackdefense.com/challengedetails?cid=1609
Type	DevSecOps : Docker Security Tools

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective: Run the tool and analyze the results!

Solution:

Step 1: Check the images present on the machine.

Command: docker images

```
root@localhost:~# docker images
REPOSITORY                TAG                 IMAGE ID            CREATED             SIZE
amicontained              latest              0abdbe6e1858        20 minutes ago      11.9MB
r.j3ss.co/amicontained    latest              0abdbe6e1858        20 minutes ago      11.9MB
ubuntu                    modified            db65a5ecad18        4 weeks ago         861MB
ubuntu                    18.04               775349758637        2 months ago        64.2MB
falco                     latest              aa9fb6ba0b5b        2 months ago        734MB
alpine                    latest              965ea09ff2eb        2 months ago        5.55MB
root@localhost:~#
```

Step 2: Run the “r.j3ss.co/amicontained” docker image to invoke amicontaned tool. The tool helps the user in visualizing the kind of privileges/capabilities/namespace mappings/runtime a container will get when run with different paraeters/arguments.

The examples usage commands are mentioned in tool github repo:
https://github.com/genuinetools/amicontained

Running a normal container

Command: docker run --rm -it r.j3ss.co/amicontained

```
root@localhost:~# docker run --rm -it r.j3ss.co/amicontained
Container Runtime: docker
Has Namespaces:
        pid: true
        user: false
AppArmor Profile: unconfined
Capabilities:
        BOUNDING -> chown dac_override fowner fsetid kill setgid setuid setpcap net_bind_service net_raw sys_chroot mknod audit_write setfcap
Seccomp: filtering
Blocked Syscalls (68):
        MSGRCV SYSLOG SETPGID SETSID USELIB USTAT SYSFS VHANGUP PIVOT_ROOT _SYSCTL ACCT SETTIMEOFDAY MOUNT UMOUNT2 SWAPON SWAPOFF REBOOT SETHOS
TNAME SETDOMAINNAME IOPL IOPERM CREATE_MODULE INIT_MODULE DELETE_MODULE GET_KERNEL_SYMS QUERY_MODULE QUOTACTL NFSSERVCTL GETPMSG PUTPMSG AFS_SY
SCALL TUXCALL SECURITY LOOKUP_DCOOKIE CLOCK_SETTIME VSERVER MBIND SET_MEMPOLICY GET_MEMPOLICY KEXEC_LOAD ADD_KEY REQUEST_KEY KEYCTL MIGRATE_PAG
ES UNSHARE MOVE_PAGES PERF_EVENT_OPEN FANOTIFY_INIT NAME_TO_HANDLE_AT OPEN_BY_HANDLE_AT CLOCK_ADJTIME SETNS PROCESS_VM_READV PROCESS_VM_WRITEV
KCMP FINIT_MODULE KEXEC_FILE_LOAD BPF USERFAULTFD MEMBARRIER PREADV2 PWRITEV2 PKEY_MPROTECT PKEY_ALLOC PKEY_FREE STATX IO_PGETEVENTS RSEQ
Looking for Docker.sock
root@localhost:~#
```

Mapping pid namespace of host to the container

Command: docker run --rm -it --pid host r.j3ss.co/amicontained

```
root@localhost:~# docker run --rm -it --pid host r.j3ss.co/amicontained
Container Runtime: docker
Has Namespaces:
        pid: false
        user: false
AppArmor Profile: unconfined
Capabilities:
        BOUNDING -> chown dac_override fowner fsetid kill setgid setuid setpcap net_bind_service net_raw sys_chroot mknod audit_write setfcap
Seccomp: filtering
Blocked Syscalls (68):
        MSGRCV SYSLOG SETPGID SETSID USELIB USTAT SYSFS VHANGUP PIVOT_ROOT _SYSCTL ACCT SETTIMEOFDAY MOUNT UMOUNT2 SWAPON SWAPOFF REBOOT SETHOS
TNAME SETDOMAINNAME IOPL IOPERM CREATE_MODULE INIT_MODULE DELETE_MODULE GET_KERNEL_SYMS QUERY_MODULE QUOTACTL NFSSERVCTL GETPMSG PUTPMSG AFS_SY
SCALL TUXCALL SECURITY LOOKUP_DCOOKIE CLOCK_SETTIME VSERVER MBIND SET_MEMPOLICY GET_MEMPOLICY KEXEC_LOAD ADD_KEY REQUEST_KEY KEYCTL MIGRATE_PAG
ES UNSHARE MOVE_PAGES PERF_EVENT_OPEN FANOTIFY_INIT NAME_TO_HANDLE_AT OPEN_BY_HANDLE_AT CLOCK_ADJTIME SETNS PROCESS_VM_READV PROCESS_VM_WRITEV
KCMP FINIT_MODULE KEXEC_FILE_LOAD BPF USERFAULTFD MEMBARRIER PREADV2 PWRITEV2 PKEY_MPROTECT PKEY_ALLOC PKEY_FREE STATX IO_PGETEVENTS RSEQ
Looking for Docker.sock
root@localhost:~#
```

Running container with apparmor=unconfined

Command: docker run --rm -it --security-opt "apparmor=unconfined" r.j3ss.co/amicontained

```
root@localhost:~# docker run --rm -it --security-opt "apparmor=unconfined" r.j3ss.co/amicontained
Container Runtime: docker
Has Namespaces:
        pid: true
        user: false
AppArmor Profile: unconfined
Capabilities:
        BOUNDING -> chown dac_override fowner fsetid kill setgid setuid setpcap net_bind_service net_raw sys_chroot mknod audit_write setfcap
Seccomp: filtering
Blocked Syscalls (68):
        MSGRCV SYSLOG SETPGID SETSID USELIB USTAT SYSFS VHANGUP PIVOT_ROOT _SYSCTL ACCT SETTIMEOFDAY MOUNT UMOUNT2 SWAPON SWAPOFF REBOOT SETHOS
TNAME SETDOMAINNAME IOPL IOPERM CREATE_MODULE INIT_MODULE DELETE_MODULE GET_KERNEL_SYMS QUERY_MODULE QUOTACTL NFSSERVCTL GETPMSG PUTPMSG AFS_SY
SCALL TUXCALL SECURITY LOOKUP_DCOOKIE CLOCK_SETTIME VSERVER MBIND SET_MEMPOLICY GET_MEMPOLICY KEXEC_LOAD ADD_KEY REQUEST_KEY KEYCTL MIGRATE_PAG
ES UNSHARE MOVE_PAGES PERF_EVENT_OPEN FANOTIFY_INIT NAME_TO_HANDLE_AT OPEN_BY_HANDLE_AT CLOCK_ADJTIME SETNS PROCESS_VM_READV PROCESS_VM_WRITEV
KCMP FINIT_MODULE KEXEC_FILE_LOAD BPF USERFAULTFD MEMBARRIER PREADV2 PWRITEV2 PKEY_MPROTECT PKEY_ALLOC PKEY_FREE STATX IO_PGETEVENTS RSEQ
Looking for Docker.sock
root@localhost:~#
```

As one can observe, amicontained tool checks the container capabilities etc and prints those for the user. This can help the user to avoid wrong command parameters while running containers.

References:

1. Docker (https://www.docker.com/)
2. Amicontained (https://github.com/genuinetools/amicontained)