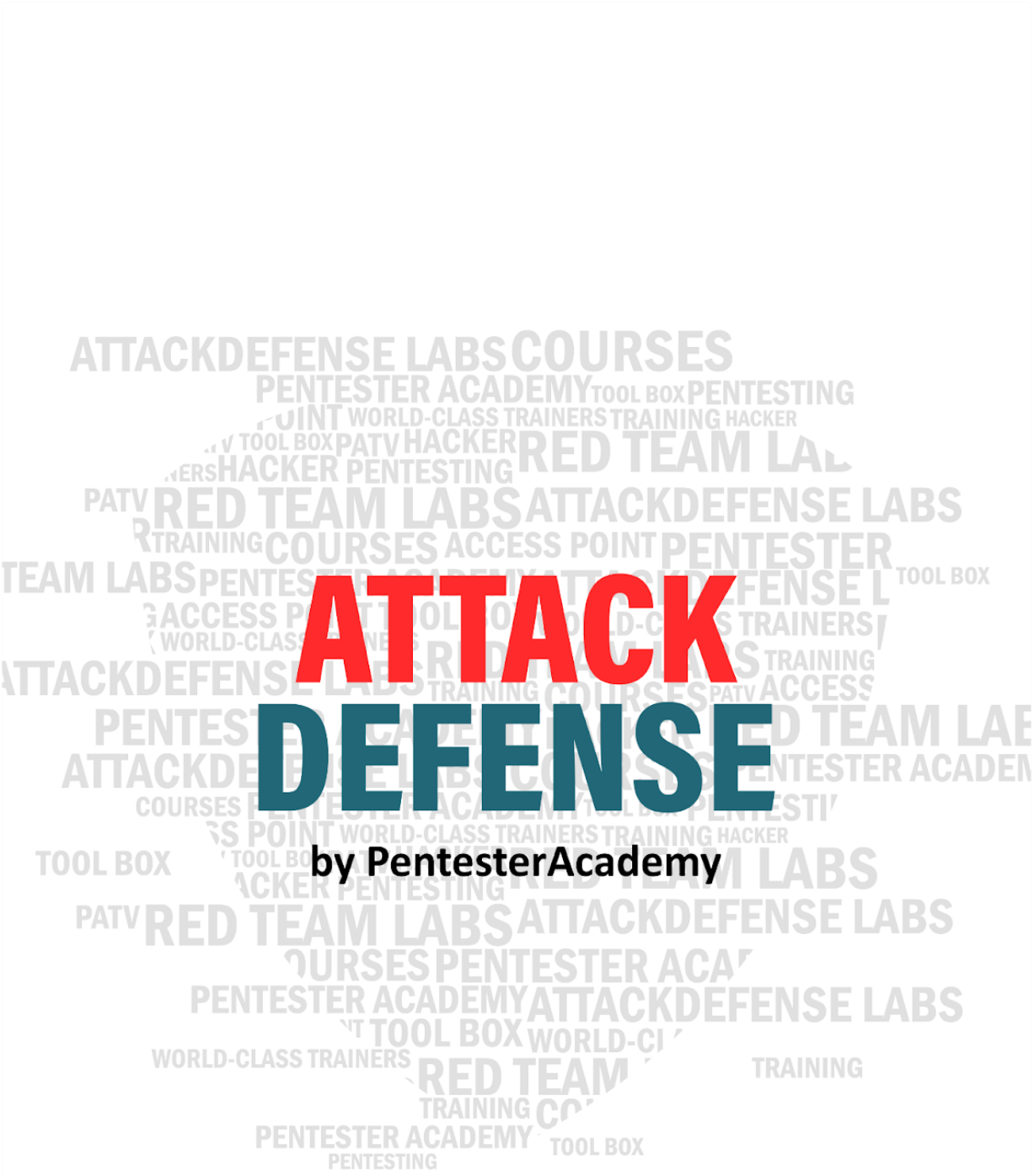
ATTACK
DEFENSE
by PentesterAcademy

Name	Falco
URL	https://attackdefense.com/challengedetails?cid=1621
Type	Docker Security : 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 Falco on the setup, launch a new container (or exec into already running container), perform malicious actions and observe the Falco logs!

Solution:

Step 1: Check the images present on the local machine.

Command: docker images

```
root@localhost:~# docker images
REPOSITORY                TAG                 IMAGE ID            CREATED             SIZE
amicontained              latest              0abdbe6e1858        3 days ago          11.9MB
r.j3ss.co/amicontained    latest              0abdbe6e1858        3 days ago          11.9MB
ubuntu                    modified            db65a5ecad18        5 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 falco image (use the command provided in the challenge guidelines).

Command: docker run -it --privileged -v /var/run/docker.sock:/host/var/run/docker.sock -v /dev:/host/dev -v /proc:/host/proc:ro -v /boot:/host/boot:ro -v /lib/modules:/host/lib/modules:ro -v /usr:/host/usr:ro falco

```
root@localhost:~# docker run -it --privileged -v /var/run/docker.sock:/host/var/run/docker.sock -v /dev:/host/dev -v /proc:/host/proc:ro -v /bo
ot:/host/boot:ro -v /lib/modules:/host/lib/modules:ro -v /usr:/host/usr:ro falco
* Setting up /usr/src links from host
* Unloading falco-probe, if present
* Running dkms install for falco
```

The container will compile the kernel module and insert it into the kernel.

```
Kernel preparation unnecessary for this kernel.  Skipping...

Building module:
cleaning build area........
make -j3 KERNELRELEASE=5.0.0-20-generic -C /lib/modules/5.0.0-20-generic/build M=/var/lib/dkms/falco/0.18.0/build
cleaning build area........

DKMS: build completed.

falco-probe.ko:
Running module version sanity check.
```

Once the kernel mode is inserted, falco loads the rules from rules directories.

```
falco-probe found and loaded in dkms
2020-01-07T06:32:56+0000: Falco initialized with configuration file /etc/falco/falco.yaml
2020-01-07T06:32:56+0000: Loading rules from file /etc/falco/falco_rules.yaml:
2020-01-07T06:32:58+0000: Loading rules from file /etc/falco/falco_rules.local.yaml:
2020-01-07T06:33:00+0000: Loading rules from file /etc/falco/k8s_audit_rules.yaml:
```

On this setup, Falco has identified 3 containers running with escalated privileges i.e. --privileged and host filesystem mount.

```
2020-01-07T06:33:03.802755000+0000: Notice Privileged container started (user=root command=container:7bf143b7ef9a stupefied_jepsen (id=7bf143b7
ef9a) image=ubuntu:18.04)
2020-01-07T06:33:03.836308000+0000: Notice Container with sensitive mount started (user=<NA> command=container:dfe6d016478e inspiring_chaum (id
=dfe6d016478e) image=ubuntu:modified mounts=/:/host::true:rslave)
2020-01-07T06:33:03.802755000+0000: Notice Privileged container started (user=root command=container:00c9e550f5a4 loving_banach (id=00c9e550f5a
4) image=falco:latest)
```

One can check the running containers and cross reference from Falco console logs to identify the issue.

```
root@localhost:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
00c9e550f5a4        falco               "/docker-entrypoint.…"   25 seconds ago      Up 17 seconds                           loving_banach
dfe6d016478e        ubuntu:modified     "supervisord -n"         7 minutes ago       Up 7 minutes                            inspiring_chaum
7bf143b7ef9a        ubuntu:18.04        "tail -f /var/log/bo…"   7 minutes ago       Up 7 minutes                            stupefied_jepsen
```

Step 3: To generate malicious activity, exec into one of the running containers.

Commands: docker exec -it 7bf143b7ef9a bash

```
root@localhost:~# docker exec -it 7bf143b7ef9a bash
root@7bf143b7ef9a:/#
root@7bf143b7ef9a:/#
```

The corresponding log alert can be observed in the Falco console logs. It clearly states that a shell was spawned in container with container_id 7bf143b7ef9a

```
2020-01-07T06:33:44.579526820+0000: Notice A shell was spawned in a container with an attached terminal (user=root stupefied_jepsen (id=7bf143b
7ef9a) shell=bash parent=runc cmdline=bash terminal=34816 container_id=7bf143b7ef9a image=ubuntu)
```

Step 4: Again, print the contents of the shadow file.

Commands: cat /etc/shadow

```
root@7bf143b7ef9a:/# cat /etc/shadow
root:*:18198:0:99999:7:::
daemon:*:18198:0:99999:7:::
bin:*:18198:0:99999:7:::
sys:*:18198:0:99999:7:::
sync:*:18198:0:99999:7:::
games:*:18198:0:99999:7:::
man:*:18198:0:99999:7:::
```

The corresponding log alert can be observed in the Falco console logs. It clearly states that a sensitive file was opened for reading in container with container_id 7bf143b7ef9a

```
2020-01-07T06:33:59.607158520+0000: Warning Sensitive file opened for reading by non-trusted program (user=root program=cat command=cat /etc/sh
adow file=/etc/shadow parent=bash gparent=<NA> ggparent=<NA> gggparent=<NA> container_id=7bf143b7ef9a image=ubuntu)
```

Step 5: Similarly, create a dummy file in /bin directory.

Commands: echo “test” > /bin/test

```
root@7bf143b7ef9a:/# echo "test" > /bin/test
root@7bf143b7ef9a:/#
```

The corresponding log alert can be observed in the Falco console logs. It clearly states that a file has been written in a known binary directory in container with container_id 7bf143b7ef9a

```
2020-01-07T06:34:26.126204400+0000: Error File below a known binary directory opened for writing (user=root command=bash file=/bin/test parent=
<NA> pcmdline=<NA> gparent=<NA> container_id=7bf143b7ef9a image=ubuntu)
```

In a similar manner, one can either rely on pre-defined rules or create custom rules to detect malicious activity.

References:

1. Docker (https://www.docker.com/)
2. Falco (https://github.com/falcosecurity/falco)