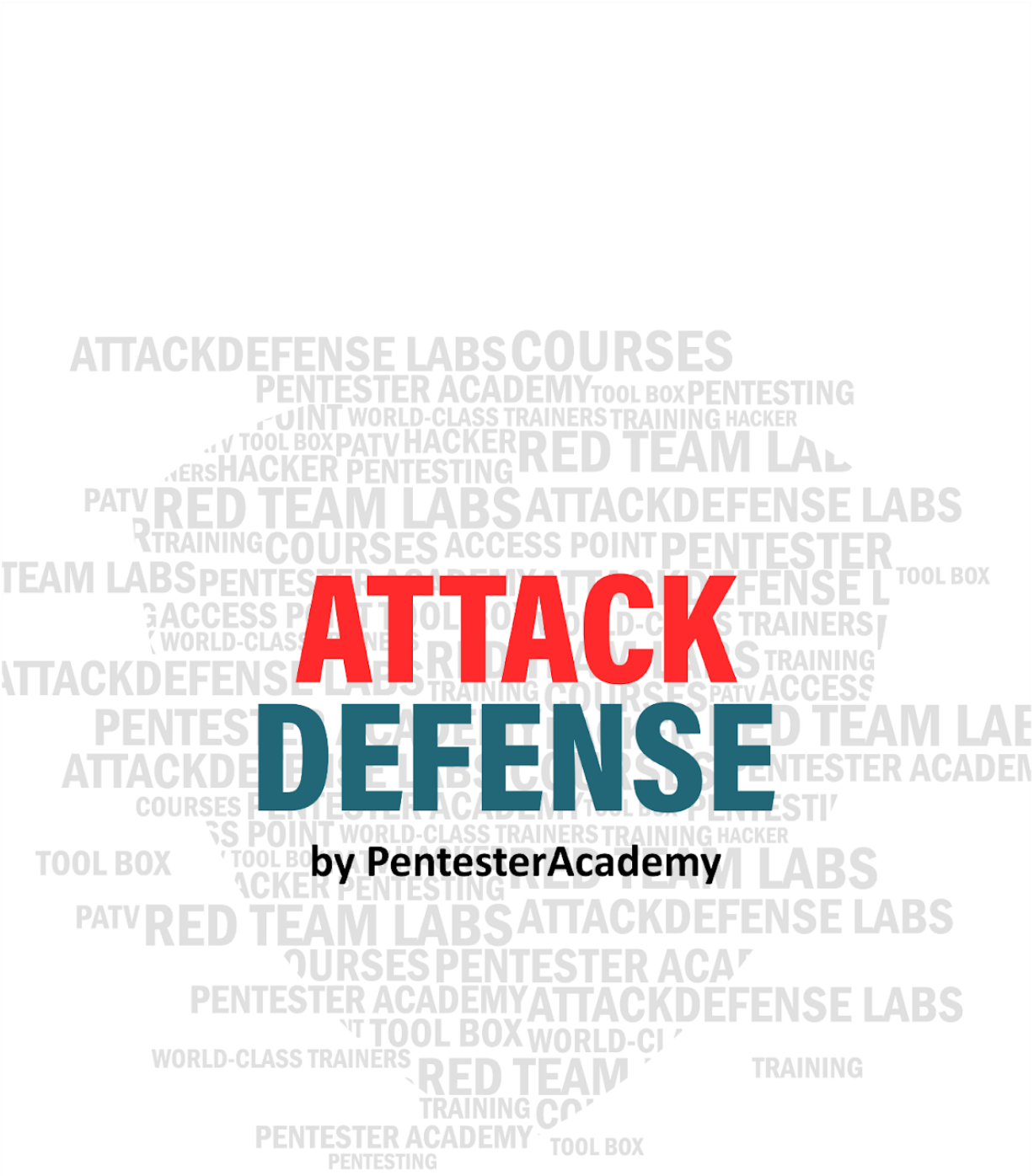
ATTACK
DEFENSE
by PentesterAcademy

Name	Flag File Forensic Recovery II
URL	https://www.attackdefense.com/challengedetails?cid=1037
Type	DevSecOps : Docker Image Forensics

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.

Step 1: Run container-diff tool on given exported image tar archive to see the image history.

Command: container-diff analyze webserver.tar --type=history

```
root@attackdefense:~# container-diff analyze webserver.tar --type=history

-----History-----

Analysis for webserver.tar:
-/bin/sh -c #(nop) ADD file:1d7cb45c4e196a6a84319b976b95ce1a9037c40b085e88350c071bf27ff59166 in /
-/bin/sh -c set -xe                && echo '#!/bin/sh' > /usr/sbin/policy-rc.d         && echo 'exit 101' >> /usr/sbin/policy-rc.d
       && chmod +x /usr/sbin/policy-rc.d                && dpkg-divert --local --rename --add /sbin/initctl         && cp -a /usr/sb
in/policy-rc.d /sbin/initctl         && sed -i 's/^exit.*/exit 0/' /sbin/initctl                && echo 'force-unsafe-io' > /etc/dpk
g/dpkg.cfg.d/docker-apt-speedup                && echo 'DPkg::Post-Invoke { "rm -f /var/cache/apt/archives/*.deb /var/cache/apt/arch
ives/partial/*.deb /var/cache/apt/*.bin || true"; };' > /etc/apt/apt.conf.d/docker-clean         && echo 'APT::Update::Post-Invoke {
"rm -f /var/cache/apt/archives/*.deb /var/cache/apt/archives/partial/*.deb /var/cache/apt/*.bin || true"; };' >> /etc/apt/apt.conf.d/
docker-clean         && echo 'Dir::Cache::pkgcache ""; Dir::Cache::srcpkgcache "";' >> /etc/apt/apt.conf.d/docker-clean
   && echo 'Acquire::Languages "none";' > /etc/apt/apt.conf.d/docker-no-languages                && echo 'Acquire::GzipIndexes "true
"; Acquire::CompressionTypes::Order:: "gz";' > /etc/apt/apt.conf.d/docker-gzip-indexes                && echo 'Apt::AutoRemove::Sugg
estsImportant "false";' > /etc/apt/apt.conf.d/docker-autoremove-suggests
-/bin/sh -c rm -rf /var/lib/apt/lists/*
-/bin/sh -c mkdir -p /run/systemd && echo 'docker' > /run/systemd/container
-/bin/sh -c #(nop)  CMD ["/bin/bash"]
-/bin/sh -c apt-get update
-/bin/sh -c apt-get install vim wget less apache2 -y
-/bin/sh -c #(nop) COPY file:7cfc3920f7692e961b5c7d5bb86caaee68ba9454ae5a3079d6ec1b8c593f2867 in /root/flag.txt
-/bin/sh -c echo "Test Web Server" > /var/www/html/index.html
-/bin/sh -c chown -R www-data:www-data /var/www/html
-/bin/sh -c #(nop) COPY file:c26512c9d423b51d20ff99e3d73e100419c3038d1bbbd6f0d185044cc7bb05fe in /root/flag.txt
-/bin/sh -c #(nop) COPY file:6b00311e9772c505f26bf1cf2406e562c32ccad58739854eea14b0855a6cb715 in /start.sh
-/bin/sh -c chmod +x /start.sh
-/bin/sh -c #(nop)  CMD ["/start.sh"]
```

Step 2: In the last step, we can observe that a file named “flag.txt” was copied to the file system. The “flag.txt” file was copied first in the 7th last layer and it was overwritten in the 4th last layer.

Command: tar -xf webserver.tar

```
root@attackdefense:~# tar -xf webserver.tar
root@attackdefense:~# ls -l
total 261096
drwxr-xr-x 2 root root      4096 May 16 08:31 2a24b751e0442b9d51fe24531fd4e748210fcac45b4abca7016729de2dbba56b
drwxr-xr-x 2 root root      4096 May 16 08:31 4ab2d03043d0b12d57899febecac0fd77e82a48161a0ef618f28e51520790bb4
drwxr-xr-x 2 root root      4096 May 16 08:31 50492564749c1a52e9d93ec015ad9d6bbc0b2ba9dc18635e5752f8d57789fd33
drwxr-xr-x 2 root root      4096 May 16 08:31 55f08f99dfd3ba11cdf9db2ab10176018657bf058f2c0f55febb10f498c887df
drwxr-xr-x 2 root root      4096 May 16 08:31 57c320d08231a2133765d63981bc9d345b7b787e9e9c1e2b1de812586f31977c
drwxr-xr-x 2 root root      4096 May 16 08:31 5fc0fb41498f287d43636892cb1a16f17494f79bb53b8c767a24c7961dce3c06
drwxr-xr-x 2 root root      4096 May 16 08:31 5fcfe5b5c9e569ded9792636cac9d7ba67bccef4f3d1c38a964bccd01b089f50
drwxr-xr-x 2 root root      4096 May 16 08:31 689df176b8419212bf4cc69c8993cfa1b47fe52bc7e6c0d65c66e0ff95aad4bf
-rw-r--r-- 1 root root      5149 May 16 08:31 6b28ba558cc6995131e7538c226dc91b3453cb5427f66e7d992e070951dd9316.json
drwxr-xr-x 2 root root      4096 May 16 08:31 70b8c620a0c5a11ba0334324524b1456d3454df23f0413ff61a4f8a60b751958
drwxr-xr-x 2 root root      4096 May 16 08:31 79509c651fdc78fab7ef4496d207b1a5e7c919ec68f32ddc658fbd07136aab95
drwxr-xr-x 2 root root      4096 May 16 08:31 bfa9206b7b7bb01e0536f4b5c43b773aae2c97199b6311b77f4c2ccd4c19f3cb
drwxr-xr-x 2 root root      4096 May 16 08:31 fb2d4623281f3a75ee458900ddb41a7e706cef3bdd0dbf80c4b91a6dfe1304a7
-rw-r--r-- 1 root root      1052 Jan  1  1970 manifest.json
-rw-r--r-- 1 root root        92 Jan  1  1970 repositories
-rw------- 1 root root 267290624 May 16 08:31 webserver.tar
root@attackdefense:~#
```

Step 3: Check manifest file to know about the ordering of the layers.

Command: cat manifest.json | python -m json.tool

```
root@attackdefense:~# cat manifest.json | python -m json.tool
[
    {
        "Config": "6b28ba558cc6995131e7538c226dc91b3453cb5427f66e7d992e070951dd9316.json",
        "Layers": [
            "70b8c620a0c5a11ba0334324524b1456d3454df23f0413ff61a4f8a60b751958/layer.tar",
            "55f08f99dfd3ba11cdf9db2ab10176018657bf058f2c0f55febb10f498c887df/layer.tar",
            "fb2d4623281f3a75ee458900ddb41a7e706cef3bdd0dbf80c4b91a6dfe1304a7/layer.tar",
            "5fcfe5b5c9e569ded9792636cac9d7ba67bccef4f3d1c38a964bccd01b089f50/layer.tar",
            "57c320d08231a2133765d63981bc9d345b7b787e9e9c1e2b1de812586f31977c/layer.tar",
            "5fc0fb41498f287d43636892cb1a16f17494f79bb53b8c767a24c7961dce3c06/layer.tar",
            "689df176b8419212bf4cc69c8993cfa1b47fe52bc7e6c0d65c66e0ff95aad4bf/layer.tar",
            "79509c651fdc78fab7ef4496d207b1a5e7c919ec68f32ddc658fbd07136aab95/layer.tar",
            "2a24b751e0442b9d51fe24531fd4e748210fcac45b4abca7016729de2dbba56b/layer.tar",
            "50492564749c1a52e9d93ec015ad9d6bbc0b2ba9dc18635e5752f8d57789fd33/layer.tar",
            "bfa9206b7b7bb01e0536f4b5c43b773aae2c97199b6311b77f4c2ccd4c19f3cb/layer.tar",
            "4ab2d03043d0b12d57899febecac0fd77e82a48161a0ef618f28e51520790bb4/layer.tar"
        ],
        "RepoTags": [
            "webserver:latest"
        ]
    }
]
root@attackdefense:~#
```

Step 4: The last layer did not create any in file system. On correlating the output of history command (step 1) with manifest file, the layer of interest seems to be 3rd from the bottom and 6th from the bottom (in manifest file listing). Extracting tar file in 3rd last layer

Command: tar -xvf 50492564749c1a52e9d93ec015ad9d6bbc0b2ba9dc18635e5752f8d57789fd33/layer.tar

```
root@attackdefense:~# tar -xvf 50492564749c1a52e9d93ec015ad9d6bbc0b2ba9dc18635e5752f8d57789fd33/layer.tar
root/
root/flag.txt
root@attackdefense:~#
```

Step 5: Viewing extracted files

Command: cat /root/flag.txt

```
root@attackdefense:~#
root@attackdefense:~# cat root/flag.txt
root@attackdefense:~#
```

Step 6: The flag.txt file extracted from the 3rd last layer was empty. Extracting 6th last layer tar file

Command: tar -xvf 689df176b8419212bf4cc69c8993cfa1b47fe52bc7e6c0d65c66e0ff95aad4bf/layer.tar

```
root@attackdefense:~# tar -xvf 689df176b8419212bf4cc69c8993cfa1b47fe52bc7e6c0d65c66e0ff95aad4bf/layer.tar
root/
root/flag.txt
root@attackdefense:~#
```

Step 7: The previous files extracted from 3rd last layer were overwritten. Viewing extracted flag.txt file.

Command: cat /root/flag.txt

```
root@attackdefense:~# cat root/flag.txt
1fa584c2f8be741abe6161119abacfb0
root@attackdefense:~#
```

Flag: 1fa584c2f8be741abe6161119abacfb0

References:

1. Docker (https://www.docker.com/)
2. Container-diff (https://github.com/GoogleContainerTools/container-diff)