Phoenix Challenges Container

In no particular order…

The following was developed from the exploit.education and other web sites (see Sources section below). The web site provides a variety of resources that can be used to learn about vulnerability analysis, exploit development, software debugging, binary analysis, and general cyber security issues.

The Phoenix virtual machine

The following description is scraped from the exploit.education web site:

```
Phoenix introduces basic memory corruption issues such as buffer overflows,
format strings and heap exploitation under "old-style" Linux system that
does not have any form of modern exploit mitigation systems enabled. It has
both 32 bit and 64 bit levels available, for both X86 and ARM systems.
Phoenix is the next progression from Nebula.
```

The files in this folder

This initial effort involves the Phoenix container because it was available as a VM (qcow2) image, where the others are available as ISO files. For those that aren't entirely familiar with Docker, I've included scripts to build the image and containers, as well as scripts to connect to the container and/or the hosted VM. In a bit more detail:

- *build-image* will build the Docker image from an Ubuntu 22.04 base image and the exploit-eduction-phoenix tarball.
- *build* will create a container from the image
- *connect-to-container* connects to the Docker container (useful when troubleshooting or if you don't have ssh installed on your host machine (instead of ssh'ing directly into the container, you can run this script to connect to the container, and then run the ssh command))
- *connect-to-vm* connects to the VM that's hosted inside of the container (basically, it invokes ssh)

Important - you won't need to run the *build-image* script unless you modify the Dockerfile.

Other files in this folder include:

- *Dockerfile* - This is the "recipe" that Docker will used to build the image
- The exploit-education tarball (the long filename, ending in ".xz") that contains the VM image for the target machine
- *startup.sh* - This is the script that the Docker image calls when the container is created. Basically, it's the qemu command that starts the VM, inside of the container.
- *notes.md* - This is the text version of this file.
- *notes.pdf* - This is *notes.md*, converted to PDF format with Pandoc. If you're reading the PDF, you're lookin' at it.

What I did to produce phoenix-container-20231127.tar.gz

```
./build-image
docker save -o phoenix-container-20231127.tar phoenix
gzip phoenix-container-20231127.tar
```

In the above, the *docker save* line exports the Docker container into an external tarball. The *gzip* line compresses the (almost) 7 GB file into a (just over) 3 GB file.

To load the tarball into Docker as an image

```
gzip -d phoenix-container-20231127.tar.gz
docker load --input phoenix-container-20231127.tar
```

In the above, the *gzip -d* line decompresses the ".tar.gz" file back into it's (almost) 7 GB file. The *docker load* line imports the file into an image called "phoenix". From there, you can run the *./build* script to deploy and start the container. Once the VM (inside of the container) starts, you can connect to it directly via:

```
ssh user@localhost -p 2222
```

If you don't have SSH installed on your host computer, you can connect to the container and from there SSH to the VM. Example:

```
./connect-to-container
ssh user@localhost -p 2222
```

Miscellaneous

- For many of the challenges, exploiting the binaries will modify the binaries. For this reason, I did not include persistent storage for the container. It is recommended that you copy each challenge before attempting to exploit it. If you mess up, exit the container, run "docker rm -f phoenix" and re-run the "./build" script. Note: you'll be back at the starting point.
- There's not a whole lot of documentation attached to this effort. It's worthwhile to reference the exploit.education web site while working the challenges. I've included a link to the challenge solutions (below).

Sources

- https://blog.lamarranet.com/index.php/exploit-education-phoenix-setup/
- https://github.com/joshkunz/qemu-docker
- https://github.com/sickcodes/Docker-OSX/issues/517

Solutions

- https://blog.lamarranet.com/index.php/exploit-education-phoenix-solutions/