Mason CC Advanced Track

Docker: Virtualization for Poor People

Recent CTFs

- Basically nothing
  - We have gone through a pretty brief dry spell
- ropchain took a team (non-Mason CC) to Catch the Ghost
  - Beat out UMBC

Upcoming CTFs & Events

- **NOT** RUSecure CTF
  - Community College and HS only
- RC3 CTF
  - Nov 17 weekend
- Idk it’s pretty chill bh

Docker & Whoops

- What is it?
  - **Containerization** software
- Whoops, not virtualization, bad clickbait is bad
  - Containerization is closer to the OS - doesn't abstract hardware or anything
- Containerization makes it cheaper and more performant - more bare metal performance
  - So poor people can run their code better
    - Total parody

Mason CC Advanced Track

Docker: ~~Virtualization~~ Containerization for Poor People

Benefits

- Command-line based
  - Instead of a shitty flash Client or something old
- Doesn't really "cost money"
  - Has EE but is basically just hosted Docker
- Expansive community
- Large amount of open source software under Docker's team
- Substantially more legal and maintainable than just slapping VMware Workstation Player, which 100% some orgs do, on a server
  - Often different use case than vSphere, vSphere lets you run Docker Swarm in vSphere
- A lot of support in cloud, orchestration, etc

Case Study

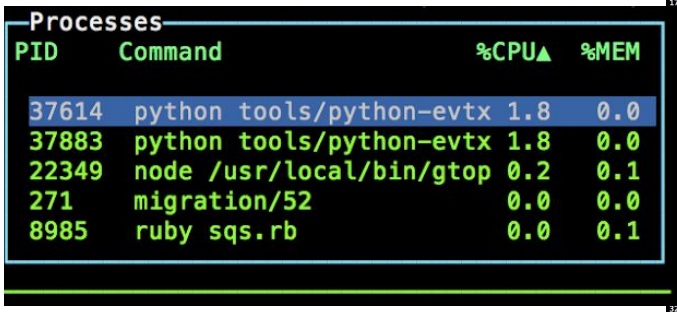
- The Crypsis Group: Crypsaurus

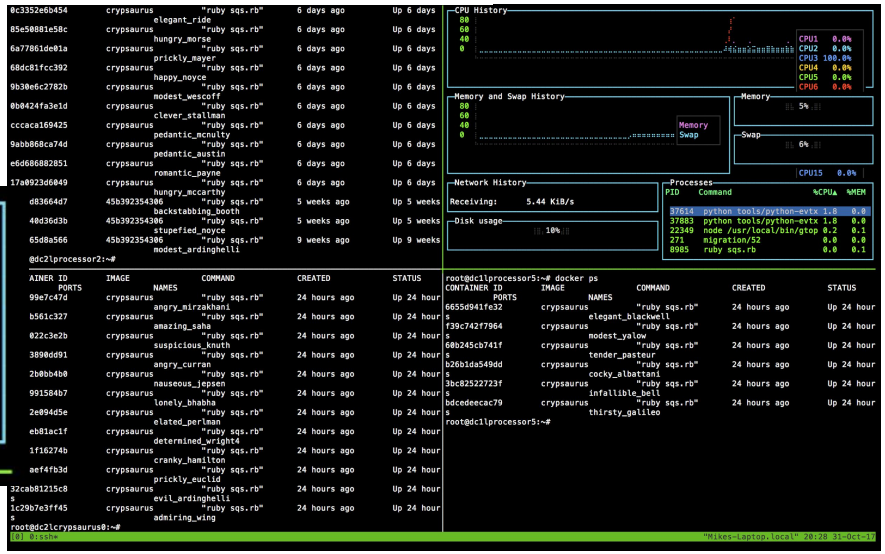
DataCruncher APP 1:15 PM
Beginning processing: 1 fa9 20171027105845-before.7z from 74 9
Beginning processing: 1 fa9 0171027111628-before.7z from 7 9

DataCruncher APP 1:45 PM
Done processing 1 -510 /1-20171027104543-after.7z

DataCruncher APP 2:17 PM
Done processing 17 -510 -20171027111628-after.7z

DataCruncher APP 2:35 PM
Done processing 17 -510aa1d3431c0a62dbf47fa9 -20171027105845-after.7z





Different Components

- Docker Engine
  - The REST API, the Daemon, and the CLI
  - The REST API is a thing by default, but bound to socket
- Docker Compose
  - WordPress example: I need a DB server and web server, compose lets you start up a DB container & web container
- Docker Swarm
  - Clustering management
  - Used for complex applications, high availability applications that may require multiple nodes, etc

K8s

- Honorable Mention: Kubernetes
  - Not under Docker, but common clustering management system comparable to Swarm
  - k8s = k u b e r n e t e s
    1 2 3 4 5 6 7 8

Why do we care about this?

- Our club tries to be well rounded….
- Portability
  - Nobody at work says “Well it worked on my system”
- Automation
  - Spin up and down different containers with ease
  - Run tests in it
- Security implications
  - Remote API is a juicy (chainable) vector
  - Host mounts is a juicy vector
  - Docker has to at least to a degree run as root
  - Frequent candidate for a sandbox
  - Heavy kernel implications, for instance bind capabilities
- TCTF runs on Docker, past challenges releasing on Docker

Installing Docker

- Docker itself has a deceiving package name:
  - docker is a system tray util in Ubuntu, so the package is **docker.io**
- Google instructions for your specific distributions, there are two common ways to install:
  - via maintainer repos
  - via Docker repos
  - Which one you do is your choice
- **Decent test: docker run -it ubuntu /bin/bash**
- Expected output:

```
mikebailey@Mikes-Laptop:~/Documents/SRCT/go|master
➜  docker run -it ubuntu /bin/bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
ae79f2514705: Pull complete
5ad56d5fc149: Pull complete
170e558760e8: Pull complete
395460e233f5: Pull complete
6f01dc62e444: Pull complete
Digest: sha256:506e2d5852de1d7c90d538c5332bd3cc33b9cbd26f6ca653875899c505c82687
Status: Downloaded newer image for ubuntu:latest
root@50325c1b0808:/#
```

Docker Hub

- Registry
- Where 99% of images tend to be
  - Generally easily auditable: Dockerfile and source usually included
- Library: kind of a group, like how a project in Gitlab would be **cc/slackbot**, CentOS would be **centos/centos7**
- Automatically pulls from by default if it can't find image locally
- There are official images, then community images, pay careful attention

Dockerfiles

- Filename **Dockerfile**
- Usually **docker build -t imgname .** in current directory
- Essentially a script to create an image
- Usually starts with base image, version and maintainer
- Has a simple command, then arguments to it
- Example node app:

```
FROM ubuntu
MAINTAINER Kimbro Staken

RUN apt-get install -y software-properties-common python
RUN add-apt-repository ppa:chris-lea/node.js
RUN echo "deb http://us.archive.ubuntu.com/ubuntu/ precise universe" >> /etc/apt/sources.list
RUN apt-get update
RUN apt-get install -y nodejs
#RUN apt-get install -y nodejs=0.6.12~dfsg1-1ubuntu1
RUN mkdir /var/www

ADD app.js /var/www/app.js

CMD ["/usr/bin/node", "/var/www/app.js"]
```

Frequent Docker CLI Commands

- **docker build** - Builds a Docker image
  - e.x. **docker build -t imagename:1.0 .**
- **docker run** - Starts a container given an image name, has many flags such as -d to run in background, -i for interactive, -t for TTY, usually used in combination with -i
- **docker stop** - stops a container via it's main PID
  - docker kill - stops container more aggressively via SIGKILL
- **docker rm** - remove container from disk
- **docker rmi** - remove image from disk
- **docker images, docker ps** - list images, **running** containers
  - **docker ps -a** lists all containers, then **docker ps -a -q** only lists container IDs
- Chaining is easy, e.x. **docker stop $(docker ps -q) && docker rm $(docker ps -a -q)**

Remote API Considerations

- **It's more explicit**
- When you just run, you run **docker run blahlibrary/image process.sh**
  - That'll pull the image, create the container and start it
- The API expects you to:
  - /images/create
  - /containers/create
  - /containers/[containerid]/start
- Runs unauthenticated by default, needs to run authenticated if bound to anything but the default socket
- REALLY fun to use in curl, as it produces CLI-like output by default for stuff like image pulling

Caching Warning

- Docker loves caching
- Docker caches images
- Docker **caches steps in a Dockerfile**
- e.x. apt update may result in old repo listings or apt install chromium may result in older Chrome
- Crypsis brought down build time from 30min to ~5min
  - Prioritized environment setup before code setup, meaning environment was cached as code changed

Image Registries (aka Registries)

- Can run it on basically anything with an open port
- Crypsis has our own
- Allows you to push out your own images to docker as docker CLI/other utilities pull from your registry rather than only Docker Hub
  - Very SSL/TLS careful

Demonstrations

- **NSA Codebreakers Challenge**
  - Register and participate in Task 0
  - Basic Docker build and run the container
  - After that has limited Docker usage
- **SRCT Go Application**
  - Good example of docker-compose
  - **git clone git.gmu.edu/srct/go** or download as zip from git.gmu.edu/srct/go
  - **docker-compose up**, note **docker ps**
- Docker Remote API Hacking
  - caffix.competitivecyber.club:1337 has Docker Remote API
  - Try to get the flag in the root of the filesystem /
    - For bonus points: add your name
- go.gmu.edu/ccvideos

Proud Sponsors

Thank you to these organizations who give us their support:

BATTELLE

It can be done™