

# The Bro Network Security Monitor



Bro Live!: Training for the Future

Jon Schipp  
NCSA

[jschipp@illinois.edu](mailto:jschipp@illinois.edu)

BroCon14  
NCSA, Champaign-Urbana, IL



# Motivations

## Issues

- ▶ **Users:** Too much time is spent passing around, downloading, and copying Virtual Machines or other materials
  - ▶ Networks are slow
  - ▶ Virtual harddisks are big
- ▶ **Users:** Technical difficulties can occur and often do that end up putting some behind the group
  - ▶ VirtualBox bus configuration
  - ▶ VirtualBox network configuration
- ▶ **Admins:** Account management is repetitive
- ▶ **Everyone:** Changes are not easy
  - ▶ Insertion of wrong exercises, mistakes, etc.. How is this handled?

⇒ Ultimately, the burden is placed on the users and this affects the overall event experience

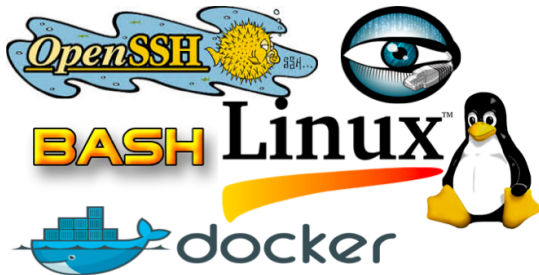
# Solutions

## Ideas

- ▶ **Admins:** Avoid passing around or downloading VM's if possible. Give user's access to your server. Big time saver!
- ▶ **Admins:** Make barrier to participation as thin as possible
  - ▶ Require only a program (e.g. ssh)
  - ▶ Opens possibilities to phones, tablets, etc.
- ▶ **Admins:** Automated account management
- ▶ **Admins:** Changes can be easily completed
  - ▶ Add, remove, or modify exercises during event
  - ▶ Immediately available

⇒ Ultimately, we pass the burden onto the admins (we're used to it anyway)

# Major Software Components

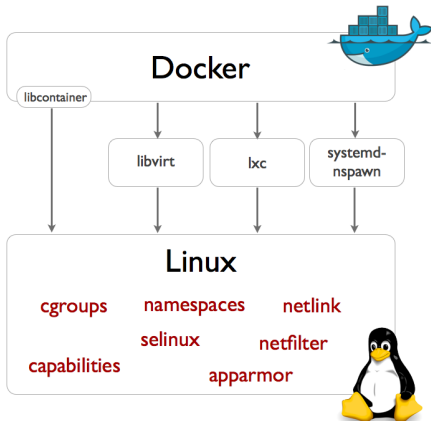


You know at least four of these right?

# Docker

## What?

- ▶ Automates the deployment of Linux based containers
- ▶ Provides a layer of abstraction
- ▶ Various methods of container creation



# Linux Based Containers

- ▶ **Important:** "Linux *Based* Containers"
  - ▶ There is no container specification
  - ▶ There are different container (and like) technologies for Linux
    - ▶ **Linux:** LXC, OpenVZ, Google containers, etc.
    - ▶ **Non-Linux:** BSD Jails, Solaris Zones, AIX WPAR, etc.
- ▶ **What** do containers do?
  - ▶ Light-weight process virtualization
- ▶ **What** do virtual machines do?
  - ▶ Hardware virtualization

# Linux Kernel Stuff

- ▶ **Support:** Linux Kernel 3.8 introduced the foundation for Linux Based containers
  - ▶ Namespaces
    - ▶ Currently available: *pid, net, ipc, uts, mnt, and user*
    - ▶ Process isolation
  - ▶ Control Groups (cgroups)
    - ▶ Resource Management
- ▶ **It's not magic**, you can create namespaces and cgroups directly from your shell by modifying `procs` and `sysfs`

# Container Advantages

- ▶ **Density:** Run hundreds or even thousands of containers on a single machine
- ▶ **Performance:** Very fast startup and tear down time, little overhead
- ▶ **Nesting:** Running containers within containers is possible
- ▶ **Isolation:** See or talk to hosts, other containers, or none
- ▶ **User Perspective:** Looks and feels like a Virtual Machine
  - ▶ Container has its own IP, filesystem, processes, etc.



# Our Implementation

1. Users log into a non-privileged system account via SSH
  - ▶ Strong crypto, ubiquitous, low overhead
  - ▶ `ssh demo@live.bro.org`
2. Automated account (non-system) creation via shell script
3. Docker is called and ships each user in their own container
  - ▶ Appropriately named and thus re-attachable by name
  - ▶ Handled via shell script
  - ▶ **Just in case you forgot each container instance is an isolated process**
4. User performs work in container
  - ▶ Runs unix commands, traverses filesystem, runs bro
5. User logs out, does something else then is ready to work again
  - 5.1 They SSH into the same non-privileged user account again
  - 5.2 Enter their newly created credentials
  - 5.3 Are automatically re-attached to their container instance

# Container Security Considerations

- ▶ Networking is disabled
  - ▶ Prevent attacks against other hosts, containers, or self
- ▶ System resources are limited per container to prevent selfishness and abuse
  - ▶ CPU and RAM allocation
- ▶ Containers and users are automatically removed after a period of time
  - ▶ Length of conference or event
- ▶ Containers which get too large are automatically removed to prevent disk space abuse
  - ▶ Denial of Service
- ▶ Finer environment controls via ulimit
  - ▶ fsize, nproc, etc.

# Want Your Own?

## You too can have one too

- ▶ Want to host your own Bro training event with a system like this?
  - ▶ It's free
  - ▶ Publicly available
    - ▶ **Vagrant:** <http://github.com/jonschipp/vagrant>
    - ▶ **Docker:** <http://hub.docker.com/u/jonschipp/latest-bro-sandbox/>
  - ▶ System configuration is entirely automated
- ▶ Written for and tested on Ubuntu Trusty and Saucy

## Installation and configuration on Ubuntu

```
$ wget https://raw.githubusercontent.com/jonschipp/vagrant/master/bro-sandbox/provision.sh - | bash
```

## Testing with Vagrant

```
$ git clone http://github.com/jonschipp/vagrant && cd  
vagrant/bro-sandbox && vagrant up; ssh -p 2222 demo@127.0.0.1
```

# Demo

## Let's try it

```
$ ssh demo@live.bro.org
demo@live.bro.org's password:
Welcome to Bro Live!
=====

...
A place to try out Bro.
Are you a new or existing user? [new/existing]: new
...
Enjoy yourself!
Training materials are located in /exercises.
e.g. $ bro -r /exercises/BroCon14/beginner/http.pcap
demo@bro: $
```

# Feedback

- ▶ **Beta:** The beta is live today!
  - ▶ Help me help you
  - ▶ Report any problems or concerns with usability or security
  - ▶ Send me feature requests
  - ▶ Send me patches and pull requests

Let me know

Talk to me

Tweet me: @JonSchipp

E-mail me: jonschipp@gmail.com, jschipp@illinois.edu

# References I



Rami Rosen

Resource management: Linux kernel Namespaces and cgroups.

In <http://www.haifux.org/lectures/299/netLec7.pdf>



Rami Rosen

Linux Containers and the Future Cloud.

In [http://www.haifux.org/lectures/320/netLec8\\_final.pdf](http://www.haifux.org/lectures/320/netLec8_final.pdf)



Jerome Petazzoni

Lightweight Virtualization with Linux Containers (LXC).

In <http://www.ciecloud.org/2013/subject/07-track06-Jerome%20Petazzoni.pdf> The 5th China Cloud Computing Conference, China National Convention Center, Beijing



Docker

[www.docker.com](http://www.docker.com)