



OPENSIFT GEARS

How to run arbitrary code on you
servers and still sleep at night

MIKE
McGrath

PRESENTER



**Mike McGrath: Operations Manager,
OpenShift Principal Architect**

AGENDA



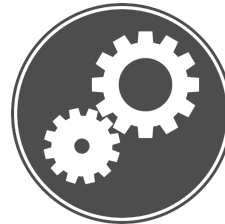
INTRO TO
OPENSIFT AND
CONTAINERS



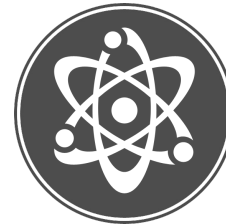
DEVS / ADMINS
WHY YOU SHOULD
CARE



DETAILS
GEARS AND
SECURITY



FUTURE
ADVANCED USES
AND ROADMAP





OPENSIFT

OpenShift is Red Hat's Platform as a Service with Origin, Online, and Enterprise



CODE

NO FORKS
COMMON
LANGUAGES
PYTHON, RUBY,
JAVA, PERL, PHP,
NODEJS, ETC.



UPLOAD

GIT INTERFACE
FULL TESTING
BUILDS
DEP RESOLUTION



RUN

MONITORING
UPDATES
SECURITY



APP CREATION

DEMO

Create a simple application against OpenShift Online for experiments later in the talk

Who Cares?

← (Devs) →

- Super low cost proof of concepts
- OpenShift online has a completely free version
- Lots of language support
- Has several built-in features like jenkins, git, etc.

← (Admins) →

- Easy place to stash applications
- Highly controlled release model
- Easy to self service in Enterprise
- High Density

Try it Free: Everyone gets 3 free gears in online, there's no reason not to try it.

INTRODUCTION TO GEARS

PRINCIPLES



UNIX USER

- In the form of a UUID
- Every user gets a UID
- All gear content is inside the users home directory



SELINUX

- Unique MCS Label
- Keyed off UID
- Determines access to system
- Protects intra-gear shenanigans



CGROUPS

- Resource Protection
- CPU limits
- Memory Limits
- Overcommit



PAM

- Using pam limits for process protections
- Using pam_namespace
- Unique /tmp/ dirs


..... **UNIX USER**

« (Good) »»


- Well understood
- Deeply tested

« (Bad) »»

- Limited resource constraints
- Security protections limited



Gear UUID: Every gear gets a unique UUID. An application consists of several gears, each with their own



..... **SELinux**

« (Good) »»

- Military grade security
- Highly customizable
- Two layers
 - First protects the system
 - Second protects the gears

« (Bad) »»

- Complicated
- Sometimes a little too good

SELinux in OpenShift: We handle it, you don't have to worry about the complicated.



- Helps keep a system online
- Keeps gear resources contained
- Keeps gear performance predictable
- Provides several accounting metrics
- Can be customized live



- Be wary of OOMs even when system memory is available



CGroups: All gears on a node have identical cgroups configurations





- Fairly well understood by senior staff
- Keeps processes contained
- :(){ :|: & }::
- Namespaces easy to setup



- Less well understood by junior staff
- Can be intrusive to non-gear accounts
- Namespaces can be confusing

PAM: namespaces in OpenShift are controlled by pam, still kernel namespaces, different user space than LXC

DEEP DIVE: SELINUX

SELINUX LABELS

s0:c6,c134 var_lib_t

SELinux uses labels to identify processes and files. openshift_t is a common process label and openshift_lib_t might be a common file label. MCS comes at the end of the label:

unconfined_u:object_r:openshift_var_lib_t:s0:c6,c134



DEEP DIVE: CGROUPS

CGROUPS

We'll look at the cgroups configurations in `/cgroups/` as well as how to view metrics and limits from within a running gear. Cgroups is useful even outside of OpenShift.

DEEP DIVE: CGROUPS

← (Memory) →

- memory.memsw.limit_in_bytes
- memory.memsw.usage_in_bytes
- memory.memsw.max_usage_in_bytes
- memory.memsw vs memory
- memory.failcnt

← (CPU) →

- cpu.cfs_period_us
- cpu.cfs_quota_us
- cpu.shares
- cpuacct.usage

← (Tasks) →

- cgroup.procs
- freezer.state (FROZEN, THAWED)

THE DEMO

LETS SEE IT

INTRODUCTION TO GEARS

PRINCIPLES (Again)



UNIX USER

- In the form of a UUID
- Every user gets a UID
- All gear content is inside the users home directory



SELINUX

- Unique MCS Label
- Keyed off UID
- Determines access to system
- Protects intra-gear shenanigans



CGROUPS

- Resource Protection
- CPU limits
- Memory Limits
- Overcommit



PAM

- Using pam limits for process protections
- Using pam_namespace
- Unique /tmp/ dirs

..... **THE FUTURE**

« (Features) »

- Custom cartridge format
- Additional Scaling Features
- Admin Console
- Stronger network separation

« (Support) »

- Integration with other layers (like IaaS)
- Additional languages
- Larger applications
- Higher performance



Questions?