



Beaker

Quick Start Guide

Introduction to Beaker
for the Impatient

Petr Šplíchal
Red Hat 2011

Abstract

- This is a short introduction to Beaker, designed for quick-learning the essential skills for automated test case writing.
- It will provide you with all the necessary steps to create a new test, while keeping the instructions as brief as possible.
- By the end you will probably realize that creating a new Beaker test is much easier than you might expect.



Beaker

The Beginning

- We start with a bug or a feature
 - A defect case or a product feature to be tested
 - Reproducer / feature test-case exists
 - The test-case can be automated
- Why write automated tests?
 - Save repeating unnecessary manual work
 - Simplify complicated setup / cleanup
 - Improve test coverage of a product
 - Prevent possible regressions

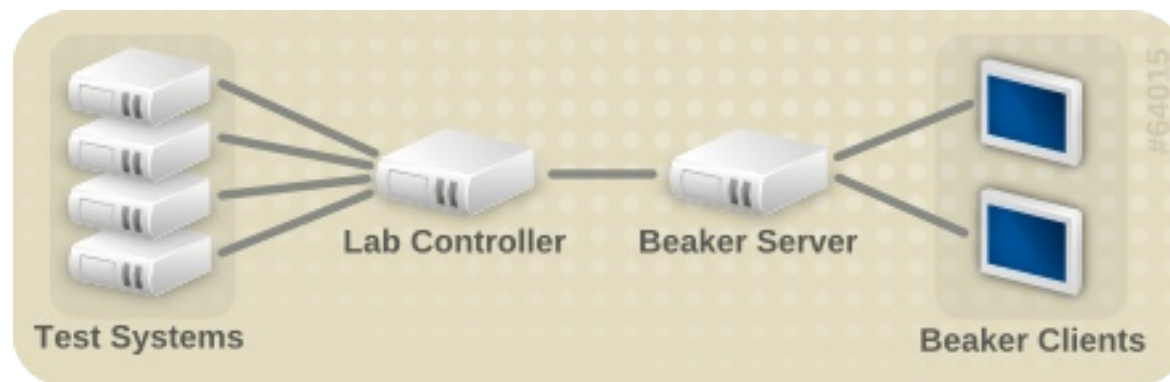
Why Beaker Tests?

- Execute same test on multiple architectures with a single command
- Build complex test-recipes (ordered testing, multihost tests, etc.)
- Execute in any language
- Large hardware inventory with easy device and system lookup capability
- Convenient BeakerLib functions
- Fully automated

Beaker Overview

■ Lab Controller

- Maintains inventory & distro data, consists of:
- Cobbler — test system interactions (distro install)
- Smolt — inventory data (test systems hardware)
- Fence-agents — power cycle (start PXE installs)
- Conserver — provides console logging



Beaker Overview

■ Beaker Server

- Central point at which all Job related activity occurs
- System inventory as well as the ability to provision Systems is also controlled from here
- Holds the repository of Tasks

■ Beaker Client

- Shell based command line interface

■ Beah Test Harness

- Responsible for executing the tasks on the system
- Currently Beah (theoretically any test harness)

Client Environment Setup

- Repository, packages and Kerberos

```
# Set up yum repo & install packages
wget -O /etc/yum.repos.d/beaker.repo http://repos.fedorapeople.org/repos/beaker\
/beaker-client-Fedora.repo
yum install -y beaker-client rhts-devel python-kerberos krb5-workstation

# Configure Beaker client (use AUTH_METHOD = "password" if not using Kerberos)
mkdir -p ~/.beaker_client
cat > ~/.beaker_client/config << EOF
HUB_URL = "https://example.com"
AUTH_METHOD = "krbv"
KRB_REALM = "EXAMPLE.COM"
EOF

# Optionally set up Kerberos
authconfig --update --enablekrb5 --krb5realm=EXAMPLE.COM \
--krb5kdc=example.com --krb5adminserver=example.com
kinit psplicha
Password for psplicha@EXAMPLE.COM: ...
```




Beaker Wizard

beaker-wizard --help

```
$ beaker-wizard --help
```

```
Usage: beaker-wizard [options] [TESTNAME] [BUG/CVE...] | beaker-wizard Makefile
```

```
Beaker Wizard is a tool which can transform that create-all-the-necessary-files-with-correct-names-values-and-paths boring phase of every test creation into one line joy. For power users there is a lot of inspiration in the extra help page. For quick start just cd to your test package directory and simply type: "beaker-wizard".
```

- Creates necessary directories & files
- Fetches bug info from Red Hat Bugzilla
- Downloads attachments / reproducers if any
- Customizable (user skeletons, defaults...)

Bug #227655 – libnet.cfg

Summary:

libnet.cfg in wrong directory

Description of problem:

The perl configuration file libnet.cfg controls whether perl CPAN requests use active or passive FTP. On x86_64 installations, this file has been placed in /usr/lib64/perl5/5.8.5/Net and has no effect on FTP. All the other libnet files are in their usual place of /usr/lib/perl5/5.8.5/Net. If this file is copied to /usr/lib/perl5/5.8.5/Net, then it is effective.

Version-Release number of selected component:

perl-5.8.5-36.RHEL4

How reproducible:

always occurs

Steps to Reproduce:

new installation of RHEL 4 on x86_64 platform

Actual results:

file /usr/lib64/perl5/5.8.5/Net/libnet.cfg is created. This has no effect.

Expected results:

this file should be /usr/lib/perl5/5.8.5/Net/libnet.cfg

Test Wizard: Running

- Enter the test directory
- Run the wizard: `beaker-wizard`
- Optionally Install the `python-bugzilla` package for Wizard's advanced features

```
# yum install -y python-bugzilla
$ mkdir -p /home/psplichka/tests/perl
$ cd /home/psplichka/tests/perl
$ beaker-wizard -by 227655
Contacting bugzilla...
Fetching details for bz227655
Examining attachments for possible reproducers
Adding test.pl (simple test using Net::Config)
Adding libnet.cfg (libnet.cfg test config file)

Ready to create the test, please review
```

Test Wizard: Review

/CoreOS/perl/Regression/bz227655-libnet-cfg-in-wrong-directory

Namespace : CoreOS
Package : perl
Test type : Regression
Relative path : None
Test name : bz227655-libnet-cfg-in-wrong-directory
Description : Test for bz227655 (libnet.cfg in wrong directory)

Bug or CVE numbers : bz227655
Reproducers to fetch : test.pl, libnet.cfg
Required packages : None
Architectures : All
Releases : All
Version : 1.0
Time : 5m

Priority : Normal
License : GPLv2
Confidential : No
Destructive : No

Skeleton : Beakerlib
Author : Petr Splichal
Email : psplicha@redhat.com

Test Wizard: Edit fields

- Change values where necessary
 - Check correct namespace, package, type
 - Pick a short / descriptive name
 - Set a reasonable time

```
[Everything OK?] namespace
```

```
Namespace
```

```
~~~~~
```

```
Possible values: distribution, kernel, desktop, tools, CoreOS, examples
```

```
[CoreOS?]
```

```
...
```

```
[Everything OK?] time
```

```
Time for test to run
```

```
~~~~~
```

```
[5m?] 10m
```

Test Wizard: Generated files

```
[Everything OK?] yes
```

```
Directory Regression/bz227655-libnet-cfg-in-wrong-directory created
```

```
File Regression/bz227655-libnet-cfg-in-wrong-directory/PURPOSE written
```

```
File Regression/bz227655-libnet-cfg-in-wrong-directory/runtest.sh written
```

```
File Regression/bz227655-libnet-cfg-in-wrong-directory/Makefile written
```

```
Attachment test.pl downloaded
```

```
Attachment libnet.cfg downloaded
```

```
$ tree
```

```
├── Regression
│   └── bz227655-libnet-cfg-in-wrong-directory
│       ├── Makefile
│       ├── PURPOSE
│       ├── libnet.cfg
│       ├── runtest.sh
│       └── test.pl
```



BeakerLib

BeakerLib: Overview

- Functions for common operations
 - Checking exit codes, managing services
 - Backup / restore, handling packages
- Journal, Phases
 - Uniform logging mechanism
 - Setup / test / cleanup phase separation
- Documentation
 - man beakerlib
 - <https://fedorahosted.org/beakerlib/wiki/Manual>

BeakerLib: Journal, Phases

■ Journal

- Logged information saved in XML format
- Easily to process / compare results
- Consistent report format
- `rlJournalStart`, `rlJournalEnd`, `rlJournalPrint`

■ The concept of phases

- `rlPhaseStart{Setup,Test,Cleanup}`, `rlPhaseEnd`
- Setup & cleanup separated from the actual test
- PASS / FAIL based on the included asserts
- Prevents false FAILs and makes debugging easier

BeakerLib: Journal, Phases

```
# runtest.sh of /CoreOS/perl/Regression/bz227655-libnet-cfg-in-wrong-directory
# Description: Test for bz227655 (libnet.cfg in wrong directory)
# Author: Petr Splichal <psplicha@redhat.com>

. /usr/bin/Beaker-environment.sh
. /usr/share/Beaker-library/Beakerlib.sh

PACKAGE="perl"

rlJournalStart
  rlPhaseStartSetup
    rlAssertRpm $PACKAGE
    rlRun "TmpDir=`mktemp -d`" 0 "Creating tmp directory"
  rlPhaseEnd

  rlPhaseStartTest
    rlAssertExists $TmpDir
    rlRun "ls -l $TmpDir" 0 "Listing tmp directory"
  rlPhaseEnd

  rlPhaseStartCleanup
    rlRun "rm -r $TmpDir" 0 "Removing tmp directory"
  rlPhaseEnd
rlJournalEnd
rlJournalPrintText
```

BeakerLib: Asserts

- Checking the exit code
 - `rlRun command [status...] [comment]` — run a command with an optional comment and make sure its exit code matches expectations
- Common checks
 - `rlAssertRpm` — make sure a package is installed
 - `rlAssertExists` — check whether a file exists
 - `rlAssertGrep` — file should contain a pattern
 - `rlAssertDiffer` — given files should differ
 - `rlAssert0`, `rlAssertEquals`, `rlAssertGreater` — arithmetic asserts used for easy comparing values

BeakerLib: Asserts

```
# runtest.sh of /CoreOS/perl/Regression/bz227655-libnet-cfg-in-wrong-directory
# Description: Test for bz227655 (libnet.cfg in wrong directory)
# Author: Petr Splichal <psplicha@redhat.com>

. /usr/bin/Beaker-environment.sh
. /usr/share/Beaker-library/Beakerlib.sh

PACKAGE="perl"

rlJournalStart
  rlPhaseStartSetup
    rlAssertRpm $PACKAGE
    rlRun "TmpDir=$(mktemp -d)" 0 "Creating tmp directory"
  rlPhaseEnd

  rlPhaseStartTest
    rlAssertExists $TmpDir
    rlRun "ls -l $TmpDir" 0 "Listing tmp directory"
  rlPhaseEnd

  rlPhaseStartCleanup
    rlRun "rm -r $TmpDir" 0 "Removing tmp directory"
  rlPhaseEnd
rlJournalEnd
rlJournalPrintText
```

BeakerLib: Services, Backup

■ Managing services

- `rlServiceStart` — make sure a service is running with fresh configuration
- `rlServiceStop` — make sure a service is stopped
- `rlServiceRestore` — restore the service into its original state

■ Backup & restore

- `rlFileBackup` — create a backup of files / directories
- `rlFileRestore` — restore backed-up files to their original location

BeakerLib: Services, Backup

```
# runtest.sh of /CoreOS/wget/Sanity/ftp
# Description: Sanity test for ftp options

FtpdConf="/etc/vsftpd/vsftpd.conf"

rlJournalStart
  rlPhaseStartSetup
    rlRun "TmpDir=\$(mktemp -d)" 0 "Creating tmp directory"
    rlRun "rlFileBackup $FtpdConf"
    rlRun "echo 'ssl_enable=NO' >> $FtpdConf" 0 "Making sure SSL is disabled"
    rlRun "rlServiceStart vsftpd"
    rlRun "useradd ftptester" 0 "Creating user ftptester"
    rlRun "pushd $TmpDir"
  rlPhaseEnd

...

  rlPhaseStartCleanup
    rlRun "popd"
    rlRun "userdel -r ftptester" 0 "Removing user ftptester"
    rlRun "rlFileRestore"
    rlRun "rlServiceRestore vsftpd"
    rlRun "rm -r $TmpDir $FtpDir" 0 "Removing test directories"
  rlPhaseEnd
rlJournalEnd
```



Test Files

Test Files: PURPOSE

- Describes what the test does
- Instructions for manual run
- Warning if destructive

```
PURPOSE of /CoreOS/perl/Regression/bz227655-libnet-cfg-in-wrong-directory
Description: Check that libnet.cfg file is placed in the correct directory
Author: Petr Splichal <psplicha@redhat.com>
Bug summary: perl configuration file libnet.cfg in wrong directory
Bugzilla link: https://bugzilla.redhat.com/show\_bug.cgi?id=227655
```

This test finds libnet.cfg file in the perl package, backs it up and replaces it with a test config file. Then checks whether the file has effect by using Net::Config module to access the config values. Finally, the libnet.cfg file is restored to its original state.

Test Files: Makefile

- Test case compilation
- Building the test package
- Submitting to the Beaker repository

```
export TEST=/CoreOS/perl/Regression/bz227655-libnet-cfg-in-wrong-directory
export TESTVERSION=1.0
```

```
FILES=$(METADATA) runtest.sh Makefile PURPOSE test.pl libnet.cfg
```

```
run: $(FILES) build
    ./runtest.sh
```

```
build: $(BUILT_FILES)
    chmod a+x ./runtest.sh test.pl
```

Test Files: Makefile / metadata

- Details for scheduling the test
- Limit architectures, releases, test time

```
$(METADATA): Makefile
  @echo "Owner:           Petr Splichal <psplicha@redhat.com>" > $(METADATA)
  @echo "Name:            $(TEST)" >> $(METADATA)
  @echo "TestVersion:     $(TESTVERSION)" >> $(METADATA)
  @echo "Path:             $(TEST_DIR)" >> $(METADATA)
  @echo "Description:      Check that libnet.cfg config file is placed in the ..."
  @echo "Type:               Regression" >> $(METADATA)
  @echo "TestTime:           10m" >> $(METADATA)
  @echo "RunFor:              perl" >> $(METADATA)
  @echo "Requires:            perl" >> $(METADATA)
  @echo "Priority:             Normal" >> $(METADATA)
  @echo "License:              GPLv2" >> $(METADATA)
  @echo "Confidential:        no" >> $(METADATA)
  @echo "Destructive:         no" >> $(METADATA)
  @echo "Bug:                  227655" >> $(METADATA)

  rhts-lint $(METADATA)
```

Test Files: test.pl and libnet.cfg

- Reproducers downloaded from Bugzilla

libnet.cfg

```
{
    'pop3_hosts' => [],
    'inet_domain' => undef,
    'time_hosts' => [],
    'smtp_hosts' => ['smtp.testing.ok'],
    ...
}
```

test.pl

```
#!/usr/bin/perl

use Net::Config qw(%NetConfig);
print %NetConfig->{smtp_hosts}[0];
```

Test Files: runtest.sh

```
PACKAGE="perl"

SmtpServer="smtp.testing.ok"
FindConfig="rpm -ql perl | grep libnet.cfg"

rlJournalStart
  rlPhaseStartSetup
    rlAssertRpm $PACKAGE
    rlRun "LibNetCfg=\${$FindConfig}" 0 "Searching for libnet.cfg file"
    rlLog "Found here: $LibNetCfg"
    rlRun "rlFileBackup $LibNetCfg" 0 "Backing up $LibNetCfg"
    rlRun "cp libnet.cfg $LibNetCfg" 0 "Copying the test libnet.cfg file"
  rlPhaseEnd

  rlPhaseStartTest
    rlRun "smtp=\$(./test.pl)" 0 "Obtaining smtp_hosts using Net::Config"
    rlLog "Net::Config says: $smtp"
    rlRun "echo $smtp | grep -q $SmtpServer" 0 "Checking for $SmtpServer"
  rlPhaseEnd

  rlPhaseStartCleanup
    rlRun "rlFileRestore" 0 "Restoring the original libnet.cfg file"
  rlPhaseEnd
rlJournalPrintText
```



Running the Test

Running the Test

- Do not run under root id unless necessary
 - rIService*, rIFile* functions require root
- Destructive operations
 - For debugging a test performing dangerous operations consider reserving a test machine
- Clean up
 - The test should leave the system in the “original” state as much as possible

```
$ make run
```

Logs: Header

```
.....  
:: [ LOG ] :: TEST PROTOCOL  
.....  
  
:: [ LOG ] :: Test run ID : debugging  
:: [ LOG ] :: Package : perl  
:: [ LOG ] :: Installed: : perl-5.8.8-24.el5.x86_64  
:: [ LOG ] :: Test started : 2009-05-19 16:35:05  
:: [ LOG ] :: Test finished : 2009-05-19 16:35:10  
:: [ LOG ] :: Test name : /CoreOS/perl/Regression/bz227655-libnet-cfg...  
:: [ LOG ] :: Distro: : Red Hat Enterprise Linux Server release 5.3  
:: [ LOG ] :: Hostname : x86-64-5s-m1.lab.bos.redhat.com  
:: [ LOG ] :: Architecture : x86_64  
  
.....  
:: [ LOG ] :: Test description  
.....  
  
PURPOSE of /CoreOS/perl/Regression/bz227655-libnet-cfg-in-wrong-directory  
Description: Check that libnet.cfg file is placed in the correct directory  
Author: Petr Splichal <psplicha@redhat.com>  
Bug summary: perl configuration file libnet.cfg in wrong directory  
Bugzilla link: https://bugzilla.redhat.com/show\_bug.cgi?id=227655
```


Logs: PASS / Setup

Test description continued...

This test finds libnet.cfg file in the perl package, backs it up and replaces it with a test config file. Then checks whether the file has effect by using Net::Config module to access the config values. Finally, the libnet.cfg file is restored to its original state.

```
.....  
:: [ LOG ] :: Setup  
.....  
  
:: [ PASS ] :: Checking for the presence of perl rpm  
:: [ PASS ] :: Searching for libnet.cfg file  
:: [ LOG ] :: Found here: /usr/lib/perl5/5.8.8/Net/libnet.cfg  
:: [ LOG ] :: Backup dir created: /tmp/Beakerlib-backup-HIN21108  
:: [ PASS ] :: Backing up /usr/lib/perl5/5.8.8/Net/libnet.cfg  
:: [ PASS ] :: Copying the test libnet.cfg file  
:: [ LOG ] :: Duration: 1s  
:: [ LOG ] :: Assertions: 4 good, 0 bad  
:: [ PASS ] :: RESULT: Setup
```

Logs: PASS / Test & Cleanup

```
.....  
:: [ LOG ] :: Test  
.....  
  
:: [ PASS ] :: Obtaining smtp_hosts using Net::Config  
:: [ LOG ] :: Net::Config says: smtp.testing.ok  
:: [ PASS ] :: Checking for smtp.testing.ok  
:: [ LOG ] :: Duration: 1s  
:: [ LOG ] :: Assertions: 2 good, 0 bad  
:: [ PASS ] :: RESULT: Test  
  
.....  
:: [ LOG ] :: Cleanup  
.....  
  
:: [ PASS ] :: Restoring the original libnet.cfg file  
:: [ LOG ] :: Duration: 0s  
:: [ LOG ] :: Assertions: 1 good, 0 bad  
:: [ PASS ] :: RESULT: Cleanup
```

Logs: FAIL / Test

```
.....  
:: [ LOG ] :: Setup  
.....  
  
:: [ PASS ] :: Checking for the presence of perl rpm  
:: [ PASS ] :: Searching for libnet.cfg file  
:: [ LOG ] :: Found here: /usr/lib64/perl5/5.8.8/Net/libnet.cfg  
:: [ LOG ] :: Backup dir created: /tmp/Beakerlib-backup-Sgb22166  
:: [ PASS ] :: Backing up /usr/lib64/perl5/5.8.8/Net/libnet.cfg  
:: [ PASS ] :: Copying the test libnet.cfg file  
:: [ LOG ] :: Duration: 1s  
:: [ LOG ] :: Assertions: 4 good, 0 bad  
:: [ PASS ] :: RESULT: Setup  
  
.....  
:: [ LOG ] :: Test  
.....  
  
:: [ PASS ] :: Obtaining smtp_hosts using Net::Config  
:: [ LOG ] :: Net::Config says:  
:: [ FAIL ] :: Checking for smtp.testing.ok (Expected 0, got 1)  
:: [ LOG ] :: Duration: 1s  
:: [ LOG ] :: Assertions: 1 good, 1 bad  
:: [ FAIL ] :: RESULT: Test
```

Logs: FAIL / Setup

- This is an example of a false FAIL
- The test was run under regular user
- FAILs in the setup phase = Abort

```
.....  
:: [ LOG ] :: Setup  
.....  
  
:: [ PASS ] :: Checking for the presence of perl rpm  
:: [ PASS ] :: Searching for libnet.cfg file  
:: [ LOG ] :: Found here: /usr/lib/perl5/5.8.8/Net/libnet.cfg  
:: [ LOG ] :: Backup dir created: /tmp/Beakerlib-backup-nEX22493  
:: [ ERROR ] :: rlFileBackup: Backup creation failed  
:: [ FAIL ] :: Backing up /usr/lib/perl5/5.8.8/Net/libnet.cfg (Expected 0, got 1)  
:: [ FAIL ] :: Copying the test libnet.cfg file (Expected 0, got 1)  
:: [ LOG ] :: Duration: 1s  
:: [ LOG ] :: Assertions: 2 good, 2 bad  
:: [ ABORT ] :: RESULT: Setup
```



Scheduling

Submitting the Test

- `make package`
 - Just build the test rpm package [optional]
- `make bkradd`
 - Build the package
 - Upload to the Beaker server

```
$ cd /home/psplicha/tests/perl/Regression/bz227655-libnet-cfg-in-wrong-directory  
$ make bkradd
```

Scheduling the Test

- Running tests across many architectures
 - Choose the tests, arch and release, submit the job
 - Email notification sent when job is completed
 - Overall result report available for review
- Web interface
 - Basic set of workflows
 - Job results, lab machines info
- Command line
 - Useful for automated scheduling from scripts
 - Other specialized workflows

Scheduling: Web

- Log in
 - Access the web interface
 - Log in with your login/password (or Kerberos ticket)
 - Inspect the Scheduler menu
- Workflows
 - Reserve — reserve a machine for manual testing
 - New job — custom job based on provided XML

Scheduling: Beaker Client

■ Beaker Client

- Command line interface to Beaker
- Available in the `beaker-client` package

■ Useful commands

- `bkr task-add` — add/update task to scheduler
- `bkr task-list` — list tasks available for distro
- `bkr task-details` — show details about task
- `bkr job-results` — get jobs/recipes results
- `bkr help` — list all available commands

Scheduling: Workflows

■ workflow-simple

- Creating basic singlehost and multihost jobs
- `--arch` — limit architectures
- `--distro` — select the desired distribution
- `--variant` — limit variant
- `--package` — run all tests for a component
- `--task` — schedule selected task(s)
- `--whiteboard` — use custom whiteboard

■ workflow-xslt

- Advanced jobs based on XSLT templates

Viewing the results

Whiteboard

```
nautilus - automation - hp-dc5700-02.rhcs.bos.redhat.com - 10 times reliability test
Provision RHEL6-6.1 [ibm-js22-08.rhcs.eng.bos.redhat.com] #115215
Provision RHEL5-Server-U7 [ibm-hs22-03.lab.bos.redhat.com.] #115210
Provision RHEL5-Server-U7 [ibm-ls22-01.rhcs.eng.brq.redhat.com] #115206
```

Filter Whiteboard

Job ID

60146
60145

Generate

Hide naks

| Task | i386 | | ia64 | | ppc64 | |
|---|---------|---------|---------|---------|---------|---------|
| | old | new | old | new | old | new |
| /CoreOS/pcr/Regression/bz457064-pcr-is-configured-with-no-support-for-Unicode | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 |
| /CoreOS/pcr/Regression/bz669413-infinite-loop-on-some-unicode-patterns | Fail: 1 | Pass: 1 | Fail: 1 | Pass: 1 | Fail: 1 | Pass: 1 |
| /CoreOS/pcr/Sanity/smoke-test | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | | |
| /CoreOS/pcr/Security/CVE-2006-7224 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 |
| /CoreOS/pcr/Security/CVE-2006-7225 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 |
| /CoreOS/pcr/Security/CVE-2006-7226 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 |
| /CoreOS/pcr/Security/CVE-2006-7228 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 |
| /CoreOS/pcr/Security/CVE-2006-7230 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 |
| /CoreOS/pcr/Security/CVE-2007-1659 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 |
| /CoreOS/pcr/Security/CVE-2007-1660 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 | Pass: 1 |



What next?

Learn more

■ Beaker

- fedorahosted.org/beaker
- fedorahosted.org/beaker/wiki/BeakerUserGuide

■ BeakerLib

- fedorahosted.org/beakerlib
- fedorahosted.org/beakerlib/wiki/Manual

■ Beaker Wizard

- fedorahosted.org/beaker/wiki/BeakerWizard



Lab Exercises

Lab

- A few simple examples to start with
- Regression
 - BZ#190539 – pbmtext crashes on x86-64
 - BZ#476551 – identify segfaults on malformed files
- Security
 - CVE-2008-1721 – python: integer signedness error
- Sanity
 - write a simple smoke test for `mod_python`

Lab Solution: 190539 - pbmtext

```
PACKAGE="netpbm"

rlJournalStart
  rlPhaseStartSetup Setup
    rlAssertRpm $PACKAGE
    rlAssertRpm "netpbm-progs"
    rlShowPackageVersion "netpbm-progs"
    rlRun "TmpDir=`mktemp -d`" 0 "Creating tmp directory"
    pushd $TmpDir
  rlPhaseEnd

  rlPhaseStartTest Testing
    rlRun "pbmtext hello > test.pbm"
    rlRun "file test.pbm | grep PBM" 0 "Checking the generated file"
  rlPhaseEnd

  rlPhaseStartCleanup Cleanup
    popd
    rlRun "rm -r $TmpDir" 0 "Removing tmp directory"
  rlPhaseEnd
rlJournalPrintText
```


Lab Solution: 476551 - identify

```
PACKAGE="ImageMagick"

rlJournalStart
  rlPhaseStartSetup Setup
    rlAssertRpm $PACKAGE
    rlRun "TmpDir=`mktemp -d`" 0 "Creating tmp directory"
    rlRun "tar xjf reproducers.tar.bz2 -C $TmpDir" 0 "Extracting images"
    pushd $TmpDir
  rlPhaseEnd

  rlPhaseStartTest Testing
    for file in *; do
      rlRun "identify $file" 0,1 "Identifying $file"
    done
  rlPhaseEnd

  rlPhaseStartCleanup Cleanup
    popd
    rlRun "rm -r $TmpDir" 0 "Removing tmp directory"
  rlPhaseEnd
rlJournalPrintText
```

Lab Solution: 442005 - python

```
$ beaker-wizard -by CVE-2008-1721
```

```
/CoreOS/python/Security/CVE-2008-1721-signedness-error-in-zlib
```

```
Namespace : CoreOS
Package   : python
Test type : Security
Test name : CVE-2008-1721-integer-signedness-error-in-zlib
Description : Test for CVE-2008-1721 (python: integer signedness error in the zlib)
Reproducers : misallocation.py, signedness.py
... : ...
```

```
PACKAGE="python"
```

```
rlJournalStart
  rlPhaseStartTest Testing
    for test in misallocation signedness; do
      rlRun "./$test.py" 1 "Testing $test"
    done
  rlPhaseEnd
rlJournalPrintText
```

Lab Solution: mod_python / 0

```
$ cd cvs/tests/mod_python
$ beaker-wizard -y smoke -d "Basic functionality test for mod_python"
```

```
Ready to create the test, please review
```

```
~~~~~
```

```
/CoreOS/mod_python/Sanity/smoke
```

```
    Namespace : CoreOS
      Package  : mod_python
    Test type  : Sanity
Relative path : None
    Test name  : smoke
  Description : Basic functionality test for mod_python

Architectures : All
    Releases  : All
    Version   : 1.0
    Time      : 5m
```

```
...
```

```
[Everything OK?] yes
```

```
Directory Sanity/smoke created
File Sanity/smoke/PURPOSE written
File Sanity/smoke/runtest.sh written
File Sanity/smoke/Makefile written
```

Lab Solution: mod_python / 1

python.conf

```
LoadModule python_module modules/mod_python.so

<Directory /var/www/html/mod-python-test>
    AddHandler python-program .py
    PythonHandler handler
    PythonDebug on
</Directory>
```

handler.py

```
#!/usr/bin/python

from mod_python import apache
import re

def handler(req):
    req.content_type = 'text/html'

    if re.search("hello.py$", req.filename): req.write("Hello World!")
    elif re.search("goodbye.py$", req.filename): req.write("Good Bye!")
    else: req.write("Requested page: " + req.filename)

    return apache.OK
```

Lab Solution: mod_python / 2

```
# runtest.sh of /CoreOS/mod_python/Sanity/smoke

PACKAGE=mod_python

# Include Beaker environment
. /usr/share/Beaker-library/Beakerlib.sh
. /usr/bin/Beaker-environment.sh

WwwDir="/var/www/html/mod-python-test"
WwwUrl="http://localhost/mod-python-test"
ModPythonConf="/etc/httpd/conf.d/python.conf"

rlJournalStart
  rlPhaseStartSetup Setup
    rlAssertRpm $PACKAGE
    rlShowPackageVersion python
    rlRun "Output=\`mktemp\`" 0 "Creating tmp file"
    rlRun "rlFileBackup $ModPythonConf"
    rlRun "cp python.conf $ModPythonConf" 0 "Creating mod_python config"
    rlRun "mkdir $WwwDir" 0 "Creating www dir"
    rlRun "cp handler.py $WwwDir" 0 "Creating handler"
    rlRun "rlServiceStart httpd"
  rlPhaseEnd

...
```

Lab Solution: mod_python / 3

...

```
rlPhaseStartTest Testing
  # hello
  rlRun "wget -O $Output $WwwUrl/hello.py" 0 "Fetching $WwwUrl/hello.py"
  rlRun "grep 'Hello World' $Output" 0 "We should get a 'Hello World' page"
  rlLog "And we got: `cat $Output`"
  # good bye
  rlRun "wget -O $Output $WwwUrl/goodbye.py" 0 "Fetching $WwwUrl/goodbye.py"
  rlRun "grep 'Good Bye' $Output" 0 "We should get a 'Good Bye' page"
  rlLog "And we got: `cat $Output`"
  # requested page
  rlRun "wget -O $Output $WwwUrl/else.py" 0 "Fetching $WwwUrl/else.py"
  rlRun "grep 'Requested page.*else.py' $Output" 0 \
    "We should get requested page name"
  rlLog "And we got: `cat $Output`"
rlPhaseEnd

rlPhaseStartCleanup Cleanup
  rlRun "rm $Output" 0 "Removing tmp file"
  rlRun "rm -r $WwwDir" 0 "Removing www dir"
  rlRun "rlFileRestore"
  rlRun "rlServiceRestore httpd"
rlPhaseEnd
rlJournalPrintText
```

Lab Solution: mod_python / 4

```
:: [ LOG ] :: Test run ID : debugging
:: [ LOG ] :: Package : mod_python
:: [ LOG ] :: Installed: : mod_python-3.3.1-8.i386
:: [ LOG ] :: Test started : 2009-09-04 14:07:05
:: [ LOG ] :: Test finished : 2009-09-04 14:07:14
:: [ LOG ] :: Test name : /CoreOS/mod_python/Sanity/smoke
:: [ LOG ] :: Distro: : Fedora release 10 (Cambridge)
:: [ LOG ] :: Hostname : localhost.localdomain
:: [ LOG ] :: Architecture : i686

...

:: [ PASS ] :: Fetching http://localhost/mod-python-test/hello.py
:: [ PASS ] :: We should get a 'Hello World' page
:: [ LOG ] :: And we got: Hello World!
:: [ PASS ] :: Fetching http://localhost/mod-python-test/goodbye.py
:: [ PASS ] :: We should get a 'Good Bye' page
:: [ LOG ] :: And we got: Good Bye!
:: [ PASS ] :: Fetching http://localhost/mod-python-test/else.py
:: [ PASS ] :: We should get requested page name
:: [ LOG ] :: And we got: Requested page: /var/www/html/mod-python-test/else.py
:: [ LOG ] :: Duration: 1s
:: [ LOG ] :: Assertions: 6 good, 0 bad
:: [ PASS ] :: RESULT: Testing
```



Questions?

The End

Thanks for listening

