

Insights Rules (cont)

Sample Rule

```
from insights import rule, make_pass, make_fail
from insights.parsers.installed_rpms import Rpm, InstalledRpms

ERROR_KEY = "BASH_BUG_PRESENT"
PASS_KEY = "BASH_BUG_NOT_PRESENT"

CONTENT = {
    ERROR_KEY: "Bash bug found! Version: {{bash}}",
    PASS_KEY: "Bash bug not found: {{bash}}."
}

@rule(InstalledRpms)
def check_bash_bug(rpms):
    bug_version = Rpm.from_package('bash-4.4.14-1.any')
    fix_version = Rpm.from_package('bash-4.4.18-1.any')
    current_version = rpms.get_max('bash')
    if bug_version <= current_version < fix_version:
        return make_fail(ERROR_KEY, bash=current_version.nvr)
    else:
        return make_pass(PASS_KEY, bash=current_version.nvr)
```

Sample Rule Test

```
from insights.core.plugins import make_pass
from insights.tests import InputData, archive_provider
from insights.specs import Specs
from eng_rules.plugins.training import bash_bug

TEST_DATA = """
{"name": "pygpgme", "epoch": "(none)", "version": "0.3", "release": "9.e17", "arch": "x86_64", ... }
{"name": "bash", "epoch": "(none)", "version": "4.2.46", "release": "31.e17", "arch": "x86_64", ... }
{"name": "rhn-check", "epoch": "(none)", "version": "2.0.2", "release": "24.e17", "arch": "x86_64", ... }
""".strip()

@archive_provider(bash_bug.check_bash_bug)
def integration_tests():
    input_data = InputData("bash_bug_older_version")
    input_data.add(Specs.installed_rpms, TEST_DATA)
    expected = make_pass(bash_bug.PASS_KEY, bash="bash-4.2.46-31.e17")
    yield input_data, expected
```