Test Plan for Red Hat Enterprise Linux version 5.6 KVM Virtualization Solution Common Criteria Certification

Owner: Debora Velarde Babb (dvelarde@us.ibm.com)

IBM Linux Technology Center – Security
15400 SW Koll Pkwy
Beaverton, OR 97006

VERIFY VERSION AND COMPLETENESS PRIOR TO USE.

Table of Contents

Chapter 1 Document control information

1.1 Change History

Revision	Date	Author	Changes
0.1	10/12/10	Debora Babb	First draft based on Test Plan for Red Hat Enterprise Linux version 5 CAPP EAL4, RBAC, and LSPP evaluation
0.2	10/27/10	Debora Babb	Updates to test case descriptions
0.3	12/14/10	Debora Babb	Removed references to additional packages needed for testing.
0.4	12/15/10	Debora Babb	Additions to Test execution section. Removed references to instal additional package Expect. Added Michael as a reviewer.
0.5	01/14/11	Debora Babb	Updates per ID review.
0.6	01/18/11	Debora Babb	Updates to test setup and execution instructions.
0.7	01/19/11	Debora Babb	Updates to test environment setup section.
0.8	01/25/11	Jim Czyzak and Debora Babb	Addition of bridge setup info to section 12.1.
0.9	02/09/11	Debora Babb	Additions and corrections to section 12.
1.0	02/09/11	Jim Czyzak and Debora Babb	Added known errors section.
1.1	02/16/11	Debora Babb	Updated per feedback from Warren, Ramon, and Jim.
1.2	02/16/11	Debora Babb	Addressed all remaining open items.
1.3	03/22/11	Debora Babb	Updated per Stephan's feedback including addition of Section 12.6.
1.4	03/23/11	Debora Babb	Added info on miscellaneous manual tests - sections 8.2 and 14.2.2.
1.5	04/14/11	Debora Babb	Added info about second expected test case failure in KVM bucket to section 13.1.

1.2 Reviewers

Name	Role

George Wilson	LTC Security Architect
Warren Grunbok	LTC Security Manager
Michael Wortman	LTC Security Project Manager
James Czyzak	Test developer
Ramon de Carvalho Valle	Test developer

Chapter 2 Availability

2.1 Availability

This document is distributed as part of the Red Hat Enterprise Linux version 5.6 KVM Virtualization Common Criteria on IBM hardware Evaluation test cases.

2.2 Completeness

Completeness of this document can be verified by checking that the “Last Page” is marked.

2.3 Obsolete copies, retention, and disposition

It is your responsibility to ensure that you have the most recent version of this document and to properly dispose of all obsolete copies.

2.4 Alternation and duplication

You may make copies of this document. You must contact the author to make changes to the document.

Chapter 3 Overview

3.1 Purpose

The purpose of the testing for this evaluation is to demonstrate the correct operation of security functions identified in the *Red Hat Enterprise Linux 5.6 KVM Security Target*. The phrase "correct operation" is defined to include appropriate failures for unauthorized or invalid access to security functions.

3.2 Scope

The test cases identified in this test plan are limited to those areas that enforce the secure operation of Red Hat Enterprise Linux 5.6. Only features and functions contained in the *Security Target* are addressed. Test cases are designed to verify the correct operation of security related user programs, databases, files, and system calls. Testing for system availability in a stress environment is beyond the scope of this plan.

Testing of alternate installation methods shall be covered by AtSec.

Chapter 4 Environment

4.1 Software and hardware

Red Hat Enterprise Linux 5.6 Server will be tested. Additional software is listed in the "Installation of test environment" section of this document.

The configuration details will be provided by the *Evaluated Configuration Guide for Red Hat Enterprise Linux 5.6 KVM on IBM hardware*. The machines must be installed according to the instructions in the *Evaluated Configuration Guide*, including required package updates. The setup of the test machines must conform strictly to the instructions and configuration details described in the *Evaluated Configuration Guide*.

4.1.1 Platforms

Final executions will be conducted on the following platforms:

Red Hat Enterprise Linux 5.6	IBM System x HS22v 64-bit	IBM System x iDataPlex dx360 M2 64-bit
Server	Yes	Yes
Client	No	No

Although tests will not be run on a client configuration, because client packages are a subset of the server packages the client code can be considered tested.

Both 32-bit and 64-bit compilation and execution will be supported and tested on all platforms using the 64-bit kernel as listed in the preceding table.

Testing will be done using the Symmetrical Multiprocessing (SMP) kernel on all platforms.

4.1.2 Additional hardware

Additional hardware might be a serial terminal or a PC with terminal emulation for some manual tests.

Chapter 5 Assumptions and dependencies

5.1 Assumptions

The following is a list of assumptions made about the testing environment.

- The majority of the test cases will execute locally to the test target machine (for example, not as a remote client). Network testing will include executions using a sending system and a receiving system, and also will include executions to localhost. SSL testing will also involve connections to a second system.
- Multiple test suites are not running concurrently on the same machine.
- The test cases have control of the execution environment. No other activity that changes system configuration can be performed simultaneously with the test cases.

5.2 Dependencies

The following dependencies are needed for successful execution of this test plan.

- The file system must be mounted with the user_xattr ACL.
- Completion and availability of the kickstart script.
- Availability of the *Evaluated Configuration Guide* and package list.
- Availability of suitable hardware.
- Availability of all software described in the Target of Evaluation (TOE), including the audit subsystem and network packages.
- Availability of sufficient test personnel.

Chapter 6 Test approach and methodology

The purpose of the test effort is to verify Red Hat Enterprise Linux 5.6 KVM Security Target compliance of Red Hat Enterprise Linux 5 and to perform some amount of Functional Verification Testing of the audit component.

The audit-test test suite (http://sourceforge.net/projects/audit-test/) was used and adapted as needed for execution on Red Hat Enterprise Linux 5.6. The test team picked up the current version of audit-test from sourceforge and will submit any necessary changes. The audit-test suite will only be run in capp mode, since lspp mode (which tests Muli-Level Security) does not apply to this evaluation.

New tests were written for areas not tested in previous evaluations.

Bug reports were written where appropriate (test cases or Red Hat code).

To count as an official run, all tests must be executed on the Red Hat Enterprise Linux 5.6 GA code with the proper configuration. Configuration details are provided by the *Evaluated Configuration Guide* included in the Evaluated Configuration rpm package.

The setup of the test machines must conform strictly to the instructions and configuration details described in the *Evaluated Configuration Guide*.

Manual testing is performed only when automated options are not available.

Test cases will be stored I CVS under the rhcc project.

Chapter 7 Target of Evaluation Compliance

The additional packages required for the test environment are all permitted according to the *Evaluated Configuration Guide (ECG).* There are no configuration violations such as setuid and setgid binaries, daemons, startup scripts, or other prohibited changes. After installation of the test environment, the system remains compliant with the Target of Evaluation (TOE).

The gcov instrumented kernel is a modified version of the TOE, and all automated gcov tests will be run to verify that behavior is identical to the TOE. The data produced by gcov will only be used to verify that internal interfaces are covered by the test suites.

Chapter 8 Test case descriptions

This section contains a short, high level descriptions of the test suites used for this evaluation.

8.1 audit-test

The HP audit-test suite will be used. Most of the audit tests are designed to be included in an automated run. There are a few tests which must be run or verified manually. The majority of the tests are in the following CVS path:

```
rhcc/HPaudit/HPaudit/audit-test/
```

8.1.1 audit-tools tests

The audit-tools tests verify correct behavior of the audit subsystems tools.

The audit-tools test_ausearch test verifies that the ausearch tool accurately locates security relevant records and exercises several of the tool's options.

The audit-tools auditd test verifies that restarting auditd results in corresponding audit records being generated.

8.1.2 audit-trail-protection test

The audit-trail-protection test verifies correct permissions of audit log files.

8.1.3 crypto tests

The crypto tests verify correct behavior and auditing of cryptographic key generation.

8.1.4 fail-safe tests

The fail-safe tests verify correct behavior of the auditd configuration settings:

- admin_space_left
- admin_space_left_action
- disk_full_action
- disk_error_action
- max_log_file
- max_log_file_action
- space_left
- space_left_action

The fail-safe tests also include test_pam_loginuid.bash which verifies that users are only allowed to login when auditd is running.

8.1.5 filter tests

The audit filter tests verify that auditing can be correctly set according to filtering rules. They are included at:

```
rhcc/HPaudit/HPaudit/audit-test/filter
```

On 64-bit systems, the audit filter tests are to be compiled and executed in both 32-bit and 64-bit modes (two passes per 64-bit platform).

8.1.6 libpam tests

The libpam tests verify correct behavior of PAM.

- test_login.bash – verifies auditing of successful logins
- test_login_fail.bash – verifies auditing of unsuccessful logins
- test_pamtally2_lock.bash – verifies that pam_tally2 will lock an account
- test_sshd.bash – verifies that successful ssh connections are audited
- test_sshd_fail.bash – verifies that unsuccessful ssh connections are audited
- test_su.bash – verifies that successful use of 'su' is audited
- test_su_fail.bash – verifies that unsuccessful use of 'su' is audited

8.1.7 network tests

The network tests verify correct behavior of different labeling protocols:

- The labeled IPSec packet labeling mechanism
- The netlabel packet labeling mechanism using CIPSO

The network tests include both IPV4 and IPV6 tests. Network tests also test both UDP and TCP.

The network tests are only run in LSPP mode, which is not covered in this evaluation. Therefore the network tests are included in the test suite but are not executed for this certification.

8.1.8 packet filtering tests

The packet filtering tests verify correct behavior of both iptables and ebtables. Because auditing is not yet enabled for both iptables and ebtables in Red Hat Enterprise Linux 5.6 and no such auditing claim is made in the Security Target, these tests do not check for corresponding audit records.

The iptables packet filtering tests are included at:
rhcc/HPaudit/HPaudit/audit-test/netfilter

The ebtables packet filtering tests are included at:

rhcc/HPaudit/HPaudit/audit-test/netfilebt

8.1.9 syscalls tests

The audit filter tests verify that the audit subsystem can correctly audit both successful and unsuccessful attempts of all security relevant syscalls. They are included at:

rhcc/HPaudit/HPaudit/audit-test/syscalls

On 64-bit systems, the audit-test/syscalls tests are to be compiled and executed in both 32-bit and 64-bit modes (two passes per 64-bit platform).

8.1.10 trusted programs tests

The trusted programs tests verify that appropriate audit records are created for success and error cases. The at and cron tests serve a dual purpose as trusted program and PAM tests. The following are tested:

- chage
- crontab
- gpasswd
- groupadd
- groupdel
- groupmod

- hwclock
- passwd
- unix_chkpwd
- useradd
- userdel
- usermod

The at test is not included in this evaluation.

8.1.11 kvm tests

The kvm tests verify correct behavior of the KVM hypervisor. Tests in the kvm bucket are named after the Security Functional Requirement (SFR) that they are intended to test. A description of all the kvm test cases is provided in the audit-test/kvm/run.conf file.

8.2 Manual tests

The idle screen lock test is the only completely manual test case. Instructions on how to manually test that the screen program correctly locks the systems after a specified time out period are included in Section 14.2.

In addition there are some partially manual tests. Those tests are included in the separate miscellaneous tarball within the manual_tests directory. Each test case includes compile and execution information within the description at the top of each file.

Chapter 9 Availability of tests, configuration guide, and test plan

After certification testing has successfully completed, a tar file will be created that includes all tests. This tar file, the configuration guide, and the test plan will be made publicly available at this location:

```
http://<link to be provided by irina>
```

The test plan will be available in the Documentation/Test Plans directory. The configuration guide will be available in the Documentation/How-To directory. The tests will be available in the Home/Project Page/Download directory.

Chapter 10 Installation of the test environment

10.1 Red Hat installation

The system must be installed with Red Hat Enterprise Linux 5.6 per the Evaluated Configuration Guide. A kickstart script is included in the Evaluated Configuration rpm file which can be used to automate much of the installation process described in the Evaluated Configuration Guide. For instructions on how to install the operating system, refer to the *Evaluated Configuration Guide for Red Hat Enterprise Linux 5.6 KVM on IBM hardware.*

10.1.1 Add a test user

A user named ealuser must be created. You will be prompted by the configuration script to create an administrative user (call it ealuser). In case ealuser is inadvertently deleted, you can recreate it by completing the following instructions:

- On the console, log in as the root user.
- Create an administrator user, as follows:

```
useradd -m -c "ealuser" -G wheel ealuser
```

- Create a password for the user according to the password policy:

```
passwd ealuser
```

- Change the user password expiration information:

```
chage -m 1 -M 60 -W 7 ealuser
```

Chapter 11 Installation of test cases

11.1 Download tests from CVS

We will maintain our own version of the HP audit-test suite in our rhcc CVS area. Our version of the HP audit-test suite includes some update that are not yet included in the official HP audit-test suite available on sourceforge.

Run the following commands to download the hp audit-test suite from our CVS area:

- **/bin/su -**
- **telnet redhat.com** (to punch through the firewall)
- **mkdir /rhcc**
- **cd /rhcc**
- **export CVS_RSH=ssh**
- **export CVSROOT=yourid@cvs.opensource.ibm.com:/cvsroot/rhcc** (Use the appropriate CVS ID instead of "yourid")
- check out the tests:

```
cvs co HPaudit/HPaudit/audit-test

OR - For anonymous access (read authority only):

cvs -z9 -d:pserver:anonymous@cvs.opensource.ibm.com:/cvsroot/rhcc co
HPaudit/HPaudit/audit-test
```

11.2 Installation with final test case tar file

Unpack the audit-test suite with the following command on the TOE and any additional network servers:

```
tar -xvjf audit-test-02112011.tar.bz2
```

Chapter 12 Test environment setup

The configuration for the networking and net filtering (iptables, ip6tables, and ebtables) tests requires setting up IP addressing for three systems:

1. TOE
2. network server - “the pitcher”
3. “the catcher”

Each system must have two ethernet devices. The network server must be running the **targeted** SELinux policy. The test cases should be installed on both the TOE and the network server. It is not necessary to have the test cases installed on “the catcher”.

12.1 Network setup

The majority of the tests will be run on a singular network. A few tests, such as forwarding, require two networks with the TOE acting as the router between two other platforms. For the routing tests we refer to the network server as “the pitcher” and the third platform (used only for the forwarding tests) as “the catcher”. The two systems referred to as “the pitcher” and “the catcher” should only have access to each other through the TOE.

The primary network device on the TOE should be the one with which has outside access for things such as ssh from a remote device. It is recommended that the secondary device be on a private network.

For the bridge testing, you will need to create a logical bridge on the TOE and add a physical interface to it. You should add the secondary physical device to the bridge you create. This should be created prior to running any of the tests in the netfilter or netfilebt directories. The bridge can be created using the following commands:

brctl addbr <bridgename>

This bridge name will be asked for by the config-server.bash script.

brctl addif <bridgename> <device>

where <device> is the device name i.e. Eth1

Next, edit the **ifcfg-<device>** file in the **/etc/sysconfig/network-scripts** directory by adding the following line:

```
BRIDGE=<bridgename>
```

If the **ifcfg-<bridgename>** file does not exist in the **/etc/sysconfig/network-scripts/** directory, create it. The IPADDR and NETMASK statements for the interface should be in the **ifcfg-<bridgename>** file.

Examples of both the ifcfg-<device> and ifcfg-<bridgename> files follow:

ifcfg-eth1 (the device file content):

```
DEVICE=eth1
HWADDR=00:14:5E:1D:E4:DA
ONBOOT=yes
BRIDGE=br1
```

ifcfg-br1 (the bridge file content):

```
DEVICE=br1
TYPE=Bridge
BOOTPROTO=static
IPADDR=192.168.1.156
NETMASK.255.255.255.0
ONBOOT=yes
```

After making the above changes, restart the network with the command:

```
    service network restart
```

12.2 The profile file

The **profile** file located in the top level directory of the audit-test suite contains a number of environment variables that must be set in order for the audit-test tests to run successfully. The environment variables can be setup manually or can be set by running a script named **config-server.bash**. The use of **config-server.bash** script is recommended for users planning to run the network or net filtering test cases. In addition to setting up environment variables, the config-server.bash script also performs other setup tasks needed for the network and net filtering test cases.

The **config-server.bash** script reads the values in **audit-test/profile** and displays those as default values when prompting the user. The **config-server.bash** script creates the file **/tmp/profile**. The environment variables can then be set by running the command: **source /tmp/profile**.

Users not using **config-server.bash**, can export the environment variables in the profile file by running the following command in the directory containing the manually created profile: **source profile**.

12.3 Environment Variables

The “profile” file located in the top level directory of the audit-test suite contains a number of environment variables that must be set in order for the audit-test tests to run successfully. Some environment variables are needed for all test cases. The majority are only needed for the network and net filtering test cases. The environment variables defined in the **profile** file are listed here.

Do not change the value of the following environment variables in the “profile” file:

PATH: Set to “PATH:.” Used to add audit-test suite top level directory to the PATH environment variable.

PPROFILE: Should be set to “capp” for this certification.

Do change the value of the following environment variables, depending on what you want to test:

MODE: 64 or 32 depending on whether test cases should be compiled into 64bit or 32bit binaries

Do change the following environment variable to match your setup before running any of the test cases:

PASSWD: Password of the privileged user running the tests.

Do change the following environment variables to match your setup before running any of the network or net filtering test cases (note that values listed here are provided as an example only):

RHOST: Usually set to “localhost”. This is the local IPv4 address of the TOE used for tests run on the loopback device.

RHOST6: Usually set to “::1”. This is the IPv6 address of the TOE used for tests run on the loopback device.

LOCAL_DEV=”eth0” The primary ethernet device on the TOE.

LOCAL_SEC_DEV=”eth1” The secondary ethernet device on the TOE.

LOCAL_IPV4="9.47.83.243" The IPv4 address of the TOE.

LOCAL_IPV6="fe80::21a:64ff:fe5c:77aa" The IPv6 address of the TOE.

LOCAL_SEC_IPV4="192.168.1.243" The secondary IPv4 address of the TOE.

LOCAL_SEC_IPV6="fe80::21a:64ff:fe5c:77ac" The secondary IPv6 address of the TOE.

LOCAL_SEC_MAC="00:1A:64:5C:77:AC/01:00:00:00:00:00" The mac address/mask of the secondary interface on the TOE.

TOE_GLOBAL=”2020::21a:64ff:fe5c:77aa" The global IPv6 address assigned to the primary ethernet device of the TOE.

TOE_SEC_GLOBAL="2010::21a:64ff:fe5c:77ac" The global IPv6 address assigned to the secondary ethernet device of the TOE.

LBLNET_SVR_IPV4="9.47.8.123" The IPv4 address of the remote platform that you will run the lblnet_tst_server program on.

LBLNET_SVR_IPV6="fe12::345:6eff:feda:7890" The IPv6 address of the remote platform that you will run the lblnet_tst_server program on.

REMOTE_IPV6_RAW="fe80::214:5eff:feda:7890" The IPv6 address of the remote platform that you will run the lblnet_tst_server program on. Should be the same value as LBLNET_SVR_IPV6.

LBLNET_SVR_DEV="eth0" The name of the ethernet device on the TOE that is used to connect to LBLNET_SVR_IPV(4 and 6) addresses.

LNET4MASK=”255.255.255.0” The IPv4 network mask used on the primary network.

LNET6MASK="64" The IPv6 network mask used on the primary network.

LBLNET_PREFIX_IPV6="fe80::/64" The IPv6 link local network prefix with mask.

SECNET_IPV4="192.4.1.0" The network address of the secondary network to which the remote network server, TOE, and the catcher server belong.

SECNET_SVR_IPV4="192.4.1.3" The IPv4 address of the secondary ethernet device of the remote server that the lblnet_tst_server program runs on.

SECNET_SVR_IPV6="fe80::214:5eff:feda:9526" The IPv6 address of the secondary ethernet device of the remote server that the lblnet_tst_server program runs on.

SECNET_SVR_DEV="eth1" The secondary device name of the remote network server.

SECNET_SVR_MAC="00:14:5E:DA:95:28/01:00:00:00:00:00"   The mac address/mask of the secondary interface on the remote network server.

SNET4MASK="255.255.255.0" The IPv4 network mask being used on the secondary network.

SNET6MASK="64" The IPv6 network mask being used on the secondary network.

CATCHER_IPV4="192.168.1.242" The IPv4 address of the third platform's ethernet device that is on the same net as LOCAL_SEC_IPV4.

CATCHER_IPV6="2010::21a:64ff:fe5c:77ac" The global IPv6 address of the third platform's ethernet device that is on the same physical network as LOCAL_SEC_IPV6 (TOE).

CATCHER_PORT4="5100" The port number that the catcher listens on in the IPv4 forwarding test.

CATCHER_PORT6="5200" The port number that the catcher listens on in the IPv6 forwarding test.

CATCHER_DEV="eth1" The secondary ethernet device of the third platform.

PITCHER_IPV6="2020::214:5eff:feda:9526" The IPv6 global address of the primary device on the platform running the lblnet_tst_server program.

PITCHER_DEV="eth0" The device name associated with the PITCHER_IPV6 address. This should be identical to LBLNET_SVR_DEV.

BRIDGE_FILTER="br1" The name of the bridge that you setup during the configuration and will be associated with and enslaved to the TOE's secondary device.

12.4 Setup needed for network and net filtering test cases

The use of the **config-server.bash** script is recommended for users planning to run the network or net filtering test cases. In addition to setting up environment variables, the **config-server.bash** script also performs other setup tasks needed for the network and net filtering test cases. This section assumes the user is using the **config-server.bash** script.

12.4.1 Commands to be run on the TOE

The following commands should be run on the TOE platform.

- Run the config-server.bash script: **audit-test/config-server.bash**
- Run the command: **source /tmp/profile**
- When asked which system you are running on, select the TOE. Continue answering the remaining questions according to your network setup.
- Copy the resulting **/tmp/profile** file to the **/tmp** directory on the network server and "the catcher" systems.

12.4.2 Commands to be run on the network server

The following commands should be run on the network server also referred to as "the pitcher".

- Verify **/tmp/profile** exists.
- Run the config-server.bash script: **audit-test/config-server.bash**
- When asked which system you are running on, select the network server. Continue answering the remaining questions according to your network setup.
- Run the command: **source /tmp/profile**
- Update the **audit-test/network/system/client_list.txt** file to match your network setup.
- Copy the edited audit-test/network/system/client_list.txt to the same directory on the TOE system.
- Setup the necessarily environment variables, by running
  - **export PPROFILE=capp**
  - **export PATH=PATH:.**
  - **export MODE=64**
  - **export PASSWD=<password of privileged user running the server program>**
- Build the test case suite which includes the network-server:
  - **cd audit-test**
  - **make**
- Start the lblnet_tst_server program by running the command:

  audit-test/utils/network-server/runserver

12.4.2 Commands to be run on "the catcher"

The following commands should be run on the third platform referred to as "the catcher".

- Verify **/tmp/profile** exists.

- Run the config-server.bash script: **audit-test/config-server.bash**
- When asked which system you are running on, select “the catcher”. Continue answering the remaining questions according to your network setup.
- Run the command: **source /tmp/profile**

12.5 Setup needed for KVM tests

The KVM tests will attempt to install guest virtual machine images. Therefore, an ISO image of an installation media that should be used to install the virtual machine environments is needed. Edit the **audit-test/kvm/config.bash** file and set the “install_media” configuration parameter to the path and file name of the ISO image to be used as the install media for the virtual machine environments.

Some test cases fail if not executed with the “SystemLow-SystemHigh” SELinux level (e.g. Logging in as a normal user and executing **su** or **sudo**). The default SELinux level for normal users is “s0”. To change the default SELinux level for normal users to “SystemLow-SystemHigh”, use the following command:

```
semanage login -m -r SystemLow-SystemHigh __default__
```

12.6 Remounts needed

If your system is installed with a separate /tmp partition, run the following remount commands:

```
mount -o remount,exec /dev/mapper/VolGroup01-temp
mount -o remount,suid /dev/mapper/VolGroup01-temp
```

Chapter 13 Known errors

13.1 Expected failures in KVM test cases

Within the KVM test bucket, two test cases are expected to fail.

One test case named **FDP_ACC.2.1.e.bash** is expected to fail. This failure has been deemed a usability problem which was not required to be corrected for this evaluation. For more details on this issue, see Red Hat bug #668458.

The second test case expected to fail is **FMT_MTD.1.1.b.bash**. This failure is also a usability problem. The current behavior is more restrictive than required, therefore change is not needed for the evaluation. For more details see Red Hat bug #684848 and related Red Hat bug #684655.

13.2 Possible failures in netfilter test bucket

The following describes a test failure that has not yet occurred, but theoretically could occur.

Several test cases where the object is to set the iptables/ip6tables target to drop for all but selected ports and where the policy for the chain is set to DROP except for selected ports have the possibility of resulting in a test error, although this behavior has yet to be experienced during development testing. The tnums of these test cases are 14,16, and 33-36. The potential cause of this is that the nc utility is integral to the audit-test network related tests. This command is used to establish the control connection between the TOE and the lblnet_tst_server which is used for passing commands from the TOE to the lblnet_tst_server. The nc command has no facility for specifying the port that tcp should acquire for the connection to the designated destination port of the lblnet_tst_server's platform. This results in the inability to predict which source port on the TOE should be designated as a destination port that ACCEPTs traffic from the 5100 port of the platform the lblnet_tst_server. Through trial and error it has been determined that the typical port range that tcp uses for this on the TOE is 30,000 – 60,000. Consequently this range is allowed on the TOE and tests messages from the lblnet_tst_server for testing the drop capability in iptables/ip6tables use destination ports outside of this range. It is possible however on some systems tcp could select a port outside of this range. If this consistently happens, the range for these tests set for iptables/ip6tables chain commands in the netfilter/run.conf file should be adjusted to compensate.

Chapter 14 Test execution

14.1 audit-test

The audit-test test suite can be loaded into any directory on the system. In the top level directory of the test suite (./audit-test/), there are a number of README files that contain information about running the tests, packages required for the tests to run, and developing new tests. The tests can be run in both capp and lspp mode. However for this evaluation, the tests should only be run in capp mode.

14.1.1 Complete setup steps

Before successfully running the audit-test suite, you must have completed the setup steps described in Section 12. Completion of the steps in Section 12 will result in a file named **profile** which is mentioned below.

14.1.2 Building audit-test

To build the 64 bit versions of the tests:

1. Change to the audit-test directory.
2. Verify that MODE is set to 64 in the file named **profile**.
3. Source the **profile** file with the command **source profile**.
4. Run the command: **make**.

To build 32bit versions of the tests:

1. Change to the audit-test directory.
2. Edit the **profile** file so that MODE=32.
3. Run the command: **make**.

14.1.3 Running audit-test

1. Login to the system as the administrative user with the following commands:

```
/bin/su -
cd <audit-test top-level directory>
source profile
```

2. To run all of the test buckets in 64-bit mode, run the following commands:

```
cd <audit-test directory>
echo $MODE (verify set to 64, if not run export MODE=64)
make run
```

3. To run all of the test buckets in 32-bit mode, run the following commands:

```
cd <audit-test directory>
echo $MODE (verify set to 32, if not run export MODE=32)
make run
```

4. To run a specific test bucket, run the following commands:

```
cd <desired test bucket directory>
echo $MODE (export MODE=32|64 if needed)
make run
```

5. To run a specific test case number within a test bucket, run the following commands:

```
cd <desired test bucket directory>
./run.bash -v <number>
```

For more information on running the audit-test suite, see **./audit-test/README.run.**

14.2 Manual tests

14.2.1 Idle screen lock

The idle screen lock test which verifies correct behavior of the screen program is manual. Steps to perform this test are:

- Login to the system as non-root user.
- Run the command: /bin/su -

- Edit /etc/screenrc. Replace “idle 600 lockscreen” with “idle 120 lockscreen”.
- Run the command: “screen”
- After 2 minutes you should be presented with a login screen asking for the non-root user's password.
- Enter non-root user's password.
- You will be prompted for root's password.
- Enter root's password. Screen should now be unlocked.

14.2.2 Miscellaneous manual tests

For instructions on how to compile and execute the manual tests included in the misc tarball, see the comments included at the top of each test case. The description of each test case includes compile and execution information.

Chapter 15 Legal notices

This work represents the view of the author and does not necessarily represent the view of IBM.

The following are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries: IBM®, eServer(TM), BladeCenter®, System x(TM), System p(TM), System i(TM), System z(TM), and z/VM®. A full list of U.S. trademarks owned by IBM may be found at this location:

```
http://www.ibm.com/legal/copytrade.shtml
```

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Red Hat and its logo are registered trademarks of Red Hat, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. Support regarding capabilities of non-IBM products should be addressed to the suppliers of those products.

Any statements about support or other commitments may be changed or cancelled at any time without notice. All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. This Information is provided "AS IS" without warranty of any kind.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice. This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk

Last Page