A Concurrent Real-Time White Paper

2881 Gateway Drive
Pompano Beach, FL 33069
(954) 974-1700
real-time.ccur.com

Real-Time Performance of a SELinux-Enabled RedHawk™ Linux® System

By: Rajiv Vaidyanath
Concurrent Consulting Engineer

March 2013
Version 1.0

Abstract

This paper is for RedHawk Linux users who are interested in understanding the real-time impact on a system with SELinux enabled. The potential overhead of common SELinux administrative activities on RedHawk's real-time determinism is also evaluated.

Real-time performance is measured when SELinux policy module loading, policy module unloading and policy violation are introduced as additional user interference on a system already under considerable I/O, memory and CPU stress.

A familiarity with RedHawk Linux System Administration and SELinux Policy Administration is assumed in this discussion.

Introduction

Security-Enhanced Linux (SELinux) is a mandatory access control mechanism that complements the discretionary access control enforcement that comes standard on Linux. The framework of SELinux is its Flask architecture. This architecture, implemented in the RedHawk Linux kernel, allows SELinux to provide flexible support for its mandatory access control policies.

RedHawk Linux supports two SELinux policy types:

- Strict
- Targeted

The **Strict** policy provides the least privilege. Access to everything is denied. The “allow” rules are required to grant privileges.

The **Targeted** policy provides protection for selected system processes, typically network daemons, while all other system processes run in *unconfined* domain. Privileges can be restricted by setting deny rules to existing policy modules. Custom policy modules can be written to enhance the security of processes not covered under the existing policy.

Enabling SELinux

There are three modes of operation for SELinux:

- Enforcing
- Disabled
- Permissive

Permissive mode logs the policy violations, but allows the actions to happen. Enforcing and Disabled are self-explanatory. Policy violation messages are generated by SELinux Access Vector Cache (AVC). These messages are logged to /var/log/messages or /var/log/audit/audit.log.

Edit /etc/sysconfig/selinux to set the appropriate mode and policy type. The auditd service is set to start-up during boot to log AVC events.

```
# chkconfig auditd on
```

For the real-time performance test, SELinux is set to *Enforcing* mode for *Targeted* policy. The system is rebooted for filesystem re-labeling to occur.

Real-Time Performance Test

The goal is to determine if enabling and performing SELinux activities on a RedHawk Linux system introduces overhead to system response time. The system chosen for this test has the following configuration:

Hardware

- TYAN S2880 motherboard
- Two AMD Opteron 244 processors
- 4GB RAM (2GB per NUMA node)

Software

- RedHawk Linux 6.3.4 (64-bit)
- ccur-rtbench package
- selinux-testsuite

The ccur-rtbench package provides the Linux RT community with standard *cyclictest* to measure real-time response time and *stress* to produce I/O, memory, CPU and disk loads. The system load is monitored via /proc/loadavg and *stress* is adjusted to provide a constant load of 50% of system capacity.

Selinux-testsuite is used to automate SELinux activities. It can be downloaded from:

```
# git clone git://git.selinuxproject.org/~serge/selinux-testsuite
```

Two scenarios are tested on the system. First with SELinux enabled in RedHawk Linux and second with SELinux disabled in RedHawk Linux. The duration of the test is 24 hours for each scenario.

RedHawk Linux with SELinux Enabled

The test system is booted with SELinux enabled. *Sestatus* confirms the setup.

```
# sestatus
  SELinux status:                 enabled
  SELinuxfs mount:                /selinux
  Current mode:                   enforcing
  Mode from config file:          enforcing
  Policy version:                 28
  Policy from config file:        targeted
```

A NUMA node is shielded for interrupts, processes, local timer interrupts and cross node memory activities and *cyclictest* is started.

```
# shield -a n1 -m1
# cyclictest -a 1 -m -p 95
```

System Load generator *stress* is started from a different xterm window.

```
# stress --cpu 20 --io 10 --vm 10
```

Loading, testing and unloading of custom SELinux policies are done repeatedly to incur consistent SELinux overhead to the RedHawk kernel. This is done by invoking the *make* commands in a shell script.

```
# cd selinux-testsuite
# make
# make -C policy load
# make -C tests test
# make -C policy unload
```

The system is also set up as a web server (hostA.ccur.com) and a policy violation is deliberately introduced by changing the context of the file it serves.

```
# ls -lZ
   -rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0 index.html

# chcon  -t home_root_t /var/www/html/index.html

# ls -lZ
   -rw-r--r--. root root system_u:object_r:home_root_t:s0 index.html

# service httpd start
```

The apache web server cannot serve up a file whose context type is not httpd_sys_content_t.

A client RedHawk Linux system runs

```
# wget hostA.ccur.com
```

in repeated one second intervals triggering AVC deny messages in hostA's /var/log/audit/audit.log:

```
type=AVC msg=audit(1361896385.781:221): avc:  denied  { getattr } for  pid=8522 comm="httpd"  path="/var/www/html/index.html"  dev="sda2"  ino=245516 scontext=unconfined_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:home_root_t:s0 tclass=file
```

After 24 hours, *cyclictest* measures a worst-case response time of **18 microseconds**.

RedHawk Linux with SELinux Disabled

Reboot the RedHawk Linux system after disabling SELinux.

```
# sestatus
  SELinux status:              disabled
```

cyclictest is run along with *stress* in the target NUMA node.

```
# shield -a n1 -m1
# cyclictest -a 1 -m -p 95
# stress --cpu 20 --io 10 --vm 10
```

After 24 hours, *cyclictest* measures a worst-case response time of **17 microseconds**.

Conclusions

Running RedHawk Linux with SELinux enabled introduces negligible overhead to real-time response time. The stability of the SELinux-enabled RedHawk Linux kernel under consistent and reliable system-wide load is robust. The future direction of SELinux is well-supported by NSA, Tresys Technology LLC and the open source community, making SELinux the security solution of choice for RedHawk Linux.

About Concurrent Real-Time

Concurrent Real-Time is the industry's foremost provider of high-performance real-time computer systems and software solutions for commercial and government markets worldwide. The company's core competencies include hardware-in-the-loop and man-in-the-loop simulation, high-speed data acquisition, process control and low-latency transactions processing for a wide range of markets. Products include the RedHawk Linux real-time operating system with guaranteed response; NightStar™ tools for advanced Linux debugging and analysis; visual imaging; and application-specific tools for simulation and testing. Concurrent (NASDAQ:CCUR) is headquartered in Atlanta, GA, with offices in North America, Europe and Asia. Concurrent Real-Time is located in Pompano Beach, FL. For more information, visit Concurrent Real-Time at www.real-time.ccur.com.