Red Hat Enterprise Linux 6 Security Feature Overview

Steve Grubb
Principal Engineer, Red Hat
June 23, 2010
Overview

- Minimal Platform Install
- Libcap-ng
- OpenSCAP
- FIPS-140
- Stronger Hashes
- Common Criteria
Minimal Platform Install

• Goals
  • Reduce Attack Surface
  • Minimize package count
  • Add back things needed for secure operation
    • Need to be able to disable services
    • Cron jobs for maintenance
    • Mail delivery for cron jobs
    • Update packages
    • Iptables, audit, and sshd
Minimal Platform Install

The default installation of Red Hat Enterprise Linux Server includes a set of software applicable for general Internet usage. What additional tasks would you like your system to include support for?

- [ ] Software Development
- [x] Virtualization
- [ ] Web server

You can further customize the software selection now, or after install via the software management application.

- [ ] Customize later
- [x] Customize now
Minimal Platform Install

Installing glibc-common-2.5-49.x86_64 (63 MB)
Common binaries and locale data for glibc
Minimal Platform Install

RHEL5 (5.5 used for testing)

- Packages - 879
- Setuid - 33
- Setgid - 11
- Daemons - 44
- Networked services - 18
- Space – 2.2 Gb
- Notes: Boots into X even though no packages checked
Minimal Platform Install

This group is a collection of graphical administration tools for the system, such as for managing user accounts and configuring system hardware.
Minimal Platform Install

RHEL5 (5.5 used for testing)

- Packages - 437
- Setuid - 29
- Setgid - 9
- Daemons - 39
- Networked services – 16
- Space – 1006 Mb
- Notes: Boots to runlevel 3
Minimal Platform Install

The default installation of Red Hat Enterprise Linux is a basic server install. You can optionally select a different set of software now.

- Virtual Host
- Desktop
- Software Development Workstation
- Minimal

Please select any additional repositories that you want to use for software installation.

- LargeFileSystem
- LoadBalance
- Red Hat Enterprise Linux

Add additional software repositories  Modify repository

You can further customize the software selection now, or after install via the software management application.

- Customize later  -  Customize now
Minimal Platform Install

Installing glibc-common-2.12-1.2.el6.x86_64 (107 MB)
Common binaries and locale data for glibc
Minimal Platform Install

RHEL6 (pre-beta2)

- Packages - 226
- Setuid - 20
- Setgid - 7
- Daemons - 13
- Networked services – 5
- Space – 565 Mb

Notes: Boots to runlevel 3 very quickly
## Minimal Platform Install - Summary

<table>
<thead>
<tr>
<th>Platform</th>
<th>Packages</th>
<th>Setuid</th>
<th>Setgid</th>
<th>Daemons</th>
<th>Network Services</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHEL5</td>
<td>879</td>
<td>33</td>
<td>11</td>
<td>44</td>
<td>18</td>
<td>2200</td>
</tr>
<tr>
<td>RHEL5 base</td>
<td>437</td>
<td>29</td>
<td>9</td>
<td>39</td>
<td>16</td>
<td>1006</td>
</tr>
<tr>
<td>RHEL6</td>
<td>226</td>
<td>20</td>
<td>7</td>
<td>13</td>
<td>5</td>
<td>565</td>
</tr>
</tbody>
</table>
Libcap-ng

- Posix Capabilities are in the Linux kernel in an attempt to make minimal privilege applications. Few examples:
  - CAP_CHOWN - this overrides the restriction of changing file ownership and group ownership.
  - CAP_NET_RAW - Allow use of RAW sockets, allow use of PACKET sockets.
  - CAP_NET_BIND_SERVICE - Allows binding to TCP/UDP sockets below 1024.
- Model consists of: Effective, Permitted, Inheritable, and Bounding Set.
Libcap-ng

- RHEL5
  - No file system based capabilities
  - Bounding set was system wide
    - Defined as !CAP_SETPCAP
    - Intended use was to prevent module loading after boot
  - Process could drop capabilities & only inheritable capabilities are passed to child processes
Libcap-ng

- RHEL6
  - Adds file system based capabilities
  - Bounding set is per thread
    - Intended use was for containers or jails
    - Threads can remove capabilities from bounding set
  - Now, processes that drop capabilities and have uid 0 can regain all capabilities on execve()
Libcap-ng

• File system based capabilities
  • You can set effective & inheritable
  • pam_cap.so can be used to add capabilities to process tree by setting permitted
  • Combining these, it's possible to construct a system that allows one user but not another be able to do certain things.
Libcap-ng

- Wanted to reduce attack surface for RHEL6
- Dropping capabilities can be used to make root daemons less powerful
- Libcap is tedious to use
  - Changing uid while retaining capabilities takes about 60 lines of code
- RHEL6 kernel has bounding set, which is not addressed by libcap
Libcap-ng

• Use Cases:
  • Drop all capabilities
  • Keep one capability
  • Keep several capabilities
  • Check if you have any capabilities
  • Check for certain capabilities
  • Retain capabilities across a uid change
Libcap-ng

- Initialize
  - Clear / Fill
  - Thread
  - File

- Modify
  - Add
  - Drop
  - Thread
  - Change UID
  - File

- Apply
Libcap-ng

- **Keep one capability**
  
  ```c
  capng_clear(CAPNG_SELECT_BOTH);
  capng_update(CAPNG_ADD, CAPNG_EFFECTIVE|CAPNG_PERMITTED, CAP_CHOWN);
  capng_apply(CAPNG_SELECT_BOTH);
  ```

- **Check if you have any capabilities**
  
  ```c
  if (capng_have_capabilities(CAPNG_SELECT_CAPS) > CAPNG_NONE)
      do_something();
  ```

- **Retain capabilities across a uid change**
  
  ```c
  capng_clear(CAPNG_SELECT_BOTH);
  capng_update(CAPNG_ADD, CAPNG_EFFECTIVE|CAPNG_PERMITTED, CAP_CHOWN);
  if (capng_change_id(99, 99, CAPNG_DROP_SUPP_GRP | CAPNG_CLEAR_BOUNDING))
      error();
  ```
Libcap-ng

• New tools to check apps:
  • Pscap – lists all applications with capabilities
  • Netcap – list all networked apps with capabilities
  • Filecap – display or set file based capabilities

• We dropped capabilities in a number of daemons to reduce the attack surface.

• We changed file permissions on important things to require CAP_DAC_OVERRIDE to write to it.
Libcap-ng

```
[root ~]# netcap
ppid  pid  acct       command          type port capabilities
1     1765  nobody     dnsmasq          tcp  53    net_admin, net_raw +
1     1652  root       sshd             tcp  22    full
1     1449  root       cupsd            tcp  631   full
1     1652  root       sshd             tcp6 22    full
1     1449  root       cupsd            tcp6 631   full
1     1449  root       cupsd            udp  631   full
1     8515  root       vpnc             udp  4500  full
1     1765  nobody     dnsmasq          udp  53    net_admin, net_raw +
1     1765  nobody     dnsmasq
```
OpenSCAP

- SCAP – Security Content Automation Protocol
- Assist users with configuring IT systems
- Used to automate:
  - Configuring systems
  - Verifying system hasn't changed
  - Verifying a vulnerability
  - Response to new threat
OpenSCAP

• Suite of Standards
  • Extensible Configuration Checklist Description Format   XCCDF
  • Open Vulnerability and Assessment Language   OVAL
  • Common Platform Enumeration   CPE
  • Common Vulnerabilities and Exposures   CVE
  • Common Configuration Enumeration   CCE
  • Common Vulnerability Scoring System   CVSS
OpenSCAP

<table>
<thead>
<tr>
<th>Question</th>
<th>Acronym</th>
</tr>
</thead>
<tbody>
<tr>
<td>What IT systems do I have in my Enterprise?</td>
<td>CPE</td>
</tr>
<tr>
<td>What vulnerabilities do I need to worry about?</td>
<td>CVE</td>
</tr>
<tr>
<td>What vulnerabilities do I need to worry about right now?</td>
<td>CVSS</td>
</tr>
<tr>
<td>How do I configure my systems securely?</td>
<td>CCE</td>
</tr>
<tr>
<td>How do I define a policy of secure configurations?</td>
<td>XCCDF</td>
</tr>
<tr>
<td>How can I be sure my systems conform to policy?</td>
<td>OVAL</td>
</tr>
</tbody>
</table>
OpenSCAP

- NSA SNAC Guide
- Baseline Security Configuration
- XCCDF Content
- New CCE assignments
- OpenSCAP Project
- Higher Level Tools
- SCAP Validation
- RHEL 6
OpenSCAP

SCAP allows the creation of text checklists as well as system reports.
OpenSCAP

- Open source library
- Free to integrate under LGPL
- Cross Platform
- Multiple languages supported
- Unicode tested
- SE Linux friendly design
- Easily extended to new platforms with plugins
OpenSCAP

- Project Goals
  - Make the standards easier to implement through open source libraries and code samples.
  - Work with tool communities to build SCAP standards and models into their offerings.

- Barriers to writing SCAP tools
  - OVAL ~400 pages
  - XCCDF 132 pages
  - Certification
OpenSCAP

Scanner

Config Tool

Vulnerability Assessment

OpenSCAP

CVSS | CPE | CCE | OVAL | XCCDF

Operating System

SUMMIT JBoss WORLD
PRESENTED BY RED HAT
OpenSCAP

- XCCDF to Kickstart
- XCCDF to Puppet
- Policy Editors
- System Integrity Scanning
  - At bootup
  - At network connect
  - During VM startup
- Adhoc query tool
- Systems Management Integration
Stronger Hashes

- MD5 was being used in many places for integrity or password hashes
- Attacks against MD5 have been getting better
- NIST's Policy on Hash Functions:
  - Federal agencies should stop using SHA-1 for digital signatures, digital time stamping and other applications that require collision resistance as soon as practical, and must use the SHA-2 family of hash functions for these applications after 2010.
  - Needed to adjust all tools that touch software from source code to system verification.
Stronger Hashes

- Shadow-utils, glibc, pam, authconfig were done during RHEL5
- Started Project for Fedora 11. Changed:
  - Rpm, koji, spacewalk/satellite, yum, createrepo, pungi, RHN, yaboot
- To do:
  - Changes for grub password hash expected in 6.1
Common Criteria

- RHEL5 was certified under LSPP at EAL4+
- No regressions in security features in RHEL6
- Challenges around protection profiles
  - NIAP – CAPP, LSPP, MRPP, GPOSPP
  - BSI - OSPP
Common Criteria

- Trusted Boot
- Integrity Verification
- Advanced Audit
- Advanced Crypto
- Labeled Security
- Virtualization
- Roles
- Directory Server Based I&A
- Base OS PP
- Trusted Boot
- Integrity Verification
- Advanced Audit
- Advanced Crypto
- Labeled Security
- Virtualization
- Roles
- Directory Server Based I&A
- Base OS PP
Common Criteria

- **Base**
  - Secure Attention Key

- **Advanced Audit**
  - Some updates regarding remote logging, performance on large files, and search by regular expression

- **Advanced Crypto**
  - Cryptography must be in separate address space from application that is using it.

- **Virtualization**
  - VM's must be separated by MAC or UID
  - Auditing: guest start/stop/pause/crash, change in resources, Qemu server accepting connections and authentication use
  - AMTU
FIPS-140

- FISMA -> SP800-53 requires FIPS certified crypto mechanism
- RHEL5
  - Data at rest: kernel (dm-crypt)
  - Data in transit: openssl, libgcrypt, nss, openssh, openswan
FIPS-140

- Libgcrypt needed strict FSM
- Integrity verification using sha256hmac
- Needed Power Up self tests in all places
- Needed Deterministic RNG in kernel
- Needed RNG test for duplicate answer
- Needed key zeroization in openssh / openswan
- Increased DSA key size for module verification
- Disallow some crypto algorithms in FIPS mode
On RHEL5, to put into FIPS-140 mode, the crypto officer must regenerate the initrd using the following command:

- `mkinitrd --with-fips -f /boot/initrd-$(uname -r).img $(uname -r)`
- Add “fips=1” to grub kernel boot line
- Reboot

To verify FIPS mode:
- `cat /proc/sys/crypto/fips_enabled`

Some other cautions in Security Policies – please read them
FIPS-140

- 2010 brings some changes (SP800-57 part1)
  - Ssh v2 protocol is no longer allowed as key distribution method
  - Diffie-Hellman key exchange must have self test
  - 112 bits of entropy required in RNG
  - Recommended key sizes almost double 1024->2048
  - Recommends some algorithms be replaced:
    - 2 key Triple DES -> 128 bit AES
    - SHA1 -> SHA2
FIPS-140

- Other crypto changes: GPOSPP, FIPS-140-3
  - Audit requirements
  - Non-debugability
  - No implementations in scripting languages
  - Separation of application and key material
Odds and Ends

- Added `pam_ssh_agent_auth` for remote use of smartcards
- Added scrub for secure disk erasing
- NetworkManager and Openswan integration
- Key Escrow system for encrypted disk partitions
Questions?

sgrubb@redhat.com
FOLLOW US ON TWITTER
www.twitter.com/redhatsummit

TWEET ABOUT IT
#summitjbw

READ THE BLOG
http://summitblog.redhat.com/