Top 10 security changes in Red Hat Enterprise Linux 8

Mark Thacker

July 2020

Product Manager, RHEL
mthacker@redhat.com

Red Hat

What we are talking about

Red Hat® Enterprise Linux® 8

- New, or new to you, features in Red Hat Enterprise Linux 8
- Provide some guidance to you about what action to take next
- Focused on the security features of Red Hat Enterprise Linux
- Not talking about ALL of the security enhancements

We aren't talking about

- The full Red Hat Enterprise Linux roadmap
- Open source community leadership
- Hardware, software, and cloud provider partnership
- **Hundreds of existing security features:**
  Common Criteria and FIPS validations, NBDE and LUKS disk crypto, AIDE, IMA, Identity Management, web SSO, etc.

Red Hat

Agenda

- Compiler flags and static code analysis
- Scanning and hardening
- Kernel live patching
- Consistent and strong crypto policy
- FIPS mode made easy
- Smart cards and HSMs
- TLS 1.3 systemwide
- Session recording
- Finer-grained SELinux support
- Trusted Platform Module usage

Red Hat

A more secure default compiler flags and static code analysis

More secure by default

Requirement for Common Criteria and other security certifications

Static code analysis performed across the entire code base

New compiler flags to prevent stack smashing and mitigate memory corruption

Guidance

- ✓ **Use the packages** that Red Hat Enterprise Linux ships.
- ✓ **Verify and examine contents** using annocheck.
- ✓ **Consider using the same defaults**, especially if you are building kernel modules.

Red Hat

Security scanning and hardening

Built-in scanning and remediation, tested by Red Hat

Scan and remediate a system with built-in OpenSCAP

- Vulnerability (CVE): using official Red Hat data
- Security configuration baseline (profile)

Available at install time to ensure secure-focused at first boot compliance

Remediate with OpenSCAP, Red Hat Ansible® Automation Platform, Red Hat Satellite, or Red Hat Insights

Security baselines include:

- PCI-DSS
- Enhanced Operating System Protection Profile (used for Common Criteria)
- Australian Cyber Security Centre (ACSC) Essential Eight
- DISA STIG

More planned as we bring over the RHEL 7 profiles to RHEL 8

Red Hat

Common Criteria certification

Market leading

Targeting every RHEL 8 extended update support (EUS) release

- 8.1 , 8.2 , 8.4 , 8.6, 8.8 - all EUS eligible

Assurance maintenance or full recertification

- 2 years certificate validity

Red Hat

Kernel live patching

Minimizes reboots for security patches

- Cumulative, critical, and important CVEs only
- 1-year “lookback”
- Supported on EUS releases only (7.7, 8.1, 8.2, 8.4, etc.)

Growing feature set over time

- User space patching of glibc, then openssl
  - Multiarch support (x64 and Power only currently)
- Enhanced Satellite integration
- Live/conventional patch alignment
- Insights integration and reporting
- Web console integration

Consistent and strong crypto policy

4 policies (with room for customization)

- Legacy
- Default
- FIPS 140
- Future

- Solves the problem of ensuring systemwide consistent cryptography settings for addressing compliance requirements
- Easy to use and easy to automate—far less error-prone

```
# update-crypto-policies --set FUTURE
# update-crypto-policies --show
```

- Sets allowed key lengths, hashes, parameters, protocols, and algorithms

Customized crypto policies

Customers can now add customizations to the existing cryptographic policies in Red Hat Enterprise Linux while retaining the consistent management and easy application.

- Create a policy from scratch (by copying an existing one)
- Create a policy delta to be applied to an existing system-supplied policy
- Applies to all crypto back ends/libraries across the whole system
- Your policy requirements, managed across the whole system

Red Hat

Systemwide effects of crypto policy

Applications and groups that follow the crypto policies

- libkrb5
- BIND
- OpenSSL
- OpenJDK
- GnuTLS
- OpenSSH
- Libreswan
- Python
- NSS

Guidance

- ✓ Use the Red Hat Enterprise Linux-provided crypto libraries and Red Hat Enterprise Linux-provided utilities
- ✓ Test with DEFAULT and FUTURE policies
- ✓ Consider using SHA256 hashes instead of SHA1

Example: Enable older TLS 1.1

LEGACY allows TLS 1.1

```
# update-crypto-policies --set LEGACY
# wget https://tls-v1-1.badssl.com:1011/
HTTP request sent, awaiting response... 200 OK
Saving to: ‘index.html’
```

DEFAULT, FIPS, and FUTURE require TLS 1.2 or better

```
# update-crypto-policies --set DEFAULT
# wget https://tls-v1-1.badssl.com:1011/
GnuTLS: A packet with illegal or unsupported version was received.
Unable to establish SSL connection.
```

TLS 1.3 systemwide

Problems

Customers requesting latest in secure networking standards

TLS 1.2 protocol being too slow for today's applications

Solutions

- TLS 1.2 redesigned (4 years in the making)
- Less clutter, faster handshake
- Modern crypto primitives (RSA-PSS, Ed25519)
- Performance: 1-RTT (0-RTT)
- Better privacy against passive observers
- Supported in OpenSSL 1.1.1, GnuTLS, and NSS

Subsystems enabled with TLS 1.3

- Apache
- GNOME
- Perl
- Python
- Ruby
- OpenJDK

Guidance

- ✓ **Update applications** to support new TLS 1.3 protocol (some differences vs. TLS 1.2)
- ✓ **Update for OpenSSL 1.1.1** (Not ABI- or API-compatible with existing OpenSSL 1.0.2)
- ✓ **OpenSSL 1.0.2** compatibility library provided, but no FIPS, no TLS 1.3

FIPS mode made easy

Less error-prone and used by all federal government customers

Enabling FIPS 140 mode in Red Hat Enterprise Linux 8

```
# fips-mode-setup --enable
# reboot
```

Guidance

- ✓ Use the Red Hat Enterprise Linux-provided crypto libraries
- ✓ Test with FIPS enabled
- ✓ FIPS validation planned for future

FIPS validation

Market leading

FIPS validation for every minor RHEL 7 and 8 release

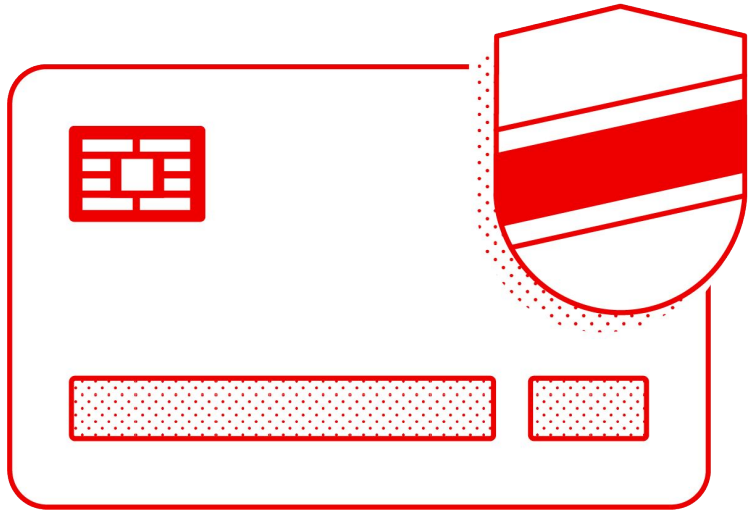
- 7.4, 7.5, 7.6, 7.7, 8.1, 8.2, 8.3, 8.4...

Yes, every minor release

- Modules not validated if unchanged in a minor release
- Includes Red Hat Enterprise Linux CoreOS

Consistent hardware security module configuration

For smart cards and hardware security modules (HSMs)



⚠ Problems

- How can my systems be hardened against Heartbleed-style attacks?
- How do I set up my smart card or HSM in Linux?
- How do I refer to an object stored in the smart card or HSM?
- How do I protect the integrity of my digital certificates, even in the cloud?

PKCS#11 centralized configuration

Smart cards and HSM devices all registered and accessed through PKCS#11

Driver registration

- Centrally via p11-kit

```
pkcs11.conf(5)
```

- Opensc is the only card driver

Using certificates

- PKCS#11 URIs:

```
pkcs11:manufacturer=piv_II;id=%01
```

OpenSSL via openssl-pkcs11

GnuTLS

NSS

Session recording

Enabling security compliance and auditing

A terminal session recording solution integrated with auditing

Records events as JSON-formatted audit records via file or syslog

Playback via terminal and web console

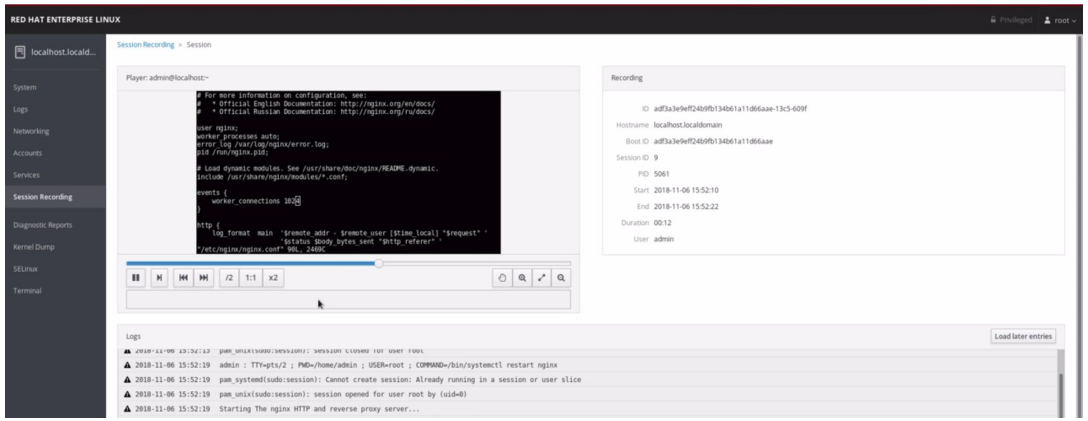
Solving the problem of recording both input and output along with environment and state of system

Selectable on a per-user, per-group basis

Session recording in action

Example

- Web console playback with actual audit events displayed in-line
- CLI playback for access from any terminal window



Guidance

- ✓ **Consider how to analyze** and use this data if you parse audit logs today
- ✓ **Consider recommending** as a security configuration in your deployment guides

Red Hat

Application control

Problems

Requirements to use only authorized applications

Need to detect or prevent modified applications from running

Solutions

Application allow list (fapolicyd)

- New, fapolicyd provides simple allow list-based control over which applications can be used
- Uses filename, hash, and path
- Small overhead for faster launching
- Predefined policy for most use cases

Integrity measurement architecture (IMA)

- Traditional solution based on hashes
- Used by Keylime and other projects and products

Fine-grained SELinux controls

⚠ Problems

Preventing inappropriate privilege escalations

- SELinux provides mandatory access control and is enabled by default
- Supports No New Privileges (NNP) in systemd `(nnp_nosuid_transition)`
- New control for preventing a process from changing the limits of another process `(getrlimit)`
- Files have specific control now to prevent certain files from being memory mapped `(file:map)`
- Ability to limit need to override access controls `(dac_read_search)`

Guidance

- ✓ Work and test with SELinux enabled—containers require it
- ✓ Review our SELinux documentation, Red Hat Summit videos, and more

Per-container SELinux policy

Problems

Reduce the need to run privileged containers

- Container security is very strict by default (each container is confined by its own SELinux label)
- Write access to certain files requires privileges (/var/log)
- Often solved by running with privileges

Reduce network access or Linux capabilities

- Processes in a container can bind to various ports
- Or use a wide variety of capabilities

Developers not SELinux experts

- May not know, or have time to learn SELinux
- May use third-party containers

Solutions

Use Udica to easily create customized policy

- Udica scans a container's configuration
- Creates a policy based on:
  - Mount points
  - Network ports
  - Capabilities
- Customized policy easily created and used
- No need to learn SELinux
- Less fuss, more security

Trusted Platform Module (TPM) usage

⚠ Problems

How to ensure integrity of the core software itself

- TPM 2.0 full support with TCG software stack
- Measurements of kernel taken each boot and stored into TPM PCR
  - No action or attestation yet, just storing the data for now
- LUKS data-at-rest key can be stored in TPM now via Network-Bound Disk Encryption utility (i.e., Clevis)
- Future work includes PKCS#11 API for TPM, virtual TPMs, and Red Hat OpenStack® Platform

Guidance

- ✓ Adopt TPM as a hardware key storage mechanism
- ✓ Interested in attestation or hardware root of trust? Look into upstream community Keylime.org

BONUS Feature!

Easier Security Management with Insights

Managing RHEL CVEs & regulatory compliance

- Insights - Web-based software as a service included with your RHEL subscription - cloud.redhat.com
- Two key services are now part of Red Hat Insights to help manage security
  - Vulnerability - help to manage, remediate, and report on RHEL CVEs
  - Compliance - easily deploy regulatory compliance policies and monitor via OpenSCAP
- Vulnerability
  - A single set of security rules provides a broader set of coverage for CVEs than before
  - Executive Reporting for at-a-glance reporting on exposures
- Compliance
  - Easy to configure and deploy OpenSCAP policies directly from within Insights Compliance service
  - Rule tailoring made easier via simple interface

Red Hat

Recap

- Compiler flags and static code analysis
- Scanning and hardening
- Kernel live patching
- Consistent and strong crypto policy
- FIPS mode made easy
- Smart cards and HSMs
- TLS 1.3 systemwide
- Session recording
- Finer-grained SELinux support
- Trusted Platform Module usage

Resources

Red Hat product security
secalert@redhat.com

Customer Portal
access.redhat.com/security

Red Hat hands-on security lab
red.ht/securitylabs

Red Hat Enterprise Linux 8
redhat.com/rhel

Red Hat

Thank you

linkedin.com/company/Red-Hat

facebook.com/RedHatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Mark Thacker
mthacker@redhat.com

Appendix of more features

Red Hat

Traditional FIPS mode enabling

Enabling FIPS 140 mode in Red Hat Enterprise Linux 7

```
# yum install dracut-fips
# yum install dracut-fips-aesni
# dracut -v -f
[Modify boot loader configuration.]
$ df /boot
$ blkid /dev/sda1
[Edit file]
# grub2-mkconfig -o /etc/grub2.cfg
# reboot
```

Very manual, not easily automated, subject to errors

Software ID (SWID) tags

⚠ Problem

How to perform software inventory management and enforce allowlisting across the enterprise

- SWID tags provide a means to consistently identify software, its origin, and manufacturer
  - Used by strongSwan, BigFix, Microsoft, and others already
- Works with any of packaging mechanisms (rpm, tar, zip, etc.)
- Defined in ISO/IEC 19770-2:2015 standard
- XML file, digitally signed by Red Hat
- Optional requirement for Common Criteria certification and required for SCAP 1.3 scanners
- Highly recommended for allowlisting for federal governments

What's next for SWID tags?

What's in Red Hat Enterprise Linux 8?

Top-level single tag identifying Red Hat Enterprise Linux itself

What's coming?

- OpenSCAP support
- Per-package tags
- Potential for non-RPM content
- Potential for container meta information
- Tools to generate SWID tags from RPM information

Guidance

- ✓ Consider delivering your own SWID tags
- ✓ Get involved in our upstream Fedora community
- ✓ Talk to us and learn more at TagVault.org

Red Hat

Examples of SWID tags

Top-level product tag in RHEL 8

```
<?xml version="1.0" encoding="utf-8"?>
<SoftwareIdentity
  xmlns="http://standards.iso.org/iso/19770/-2/2015/schema.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://standards.iso.org/iso/19770/-2/2015/schema.xsd http
://standards.iso.org/iso/19770/-2/2015-current/schema.xsd"
  xml:lang="en-US"
  name="Red Hat Enterprise Linux"
  tagId="com.redhat.RHEL-8-x86_64"
  tagVersion="1"
  version="8"
  versionScheme="multipartnumeric"
  media="(OS:linux)">
...
```

Libssh: The library for SSH communications

⚠ Problems

Applications need programmatic access to remote systems

- SSH is the de facto remote access protocol
- Applications need to contact remote systemd (web console, curl, qemu)
- The OpenSSH client application does not fit all needs
- Libssh is FIPS 140-2 compliant
- Libssh was previously in Red Hat Enterprise Linux 7 extras and now is in core Red Hat Enterprise Linux 8

Guidance

- ✓ Use libssh for remote access to systems from within your applications
- ✓ Use the system-supplied crypto libraries (notice a trend yet?)

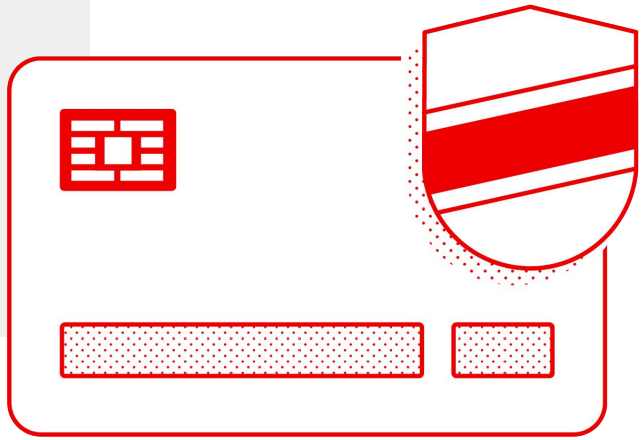
Example: Smart cards with OpenSSH

Use OpenSSH with smart card on Red Hat Enterprise Linux 8

```
$ ssh -i ‘pkcs11:id=%10’ ssh.example.com
Enter PIN for 'SSH key':

$ wget https://www.example.com/ --certificate ‘pkcs11:id=%10’
--private-key ‘pkcs11:id=%10’

$ curl https://www.example.com/ -E ‘pkcs11:id=%10;type=cert’
   --key ‘pkcs11:id=%10;type=private?pin-value=XXXX’
```



Example: HSM with Apache web server

How do I set up Apache HTTPD with an HSM on Red Hat Enterprise Linux 8?

How to set up

As simple as replacing file names with PKCS#11 URIs in the Apache configuration

```
SSLCertificateKeyFile"pkcs11:token=My%20To
en%20Name;id=45?pin-value=XXXX"

SSLCertificateFile"pkcs11:token=My%20Token%
20Name;id=45"
```

Guidance

- ✓ Use a PKCS#11 plug-in for your HSM or crypto device to work with Red Hat Enterprise Linux 8
- ✓ Especially important if you access a cloud-based HSM

Red Hat

Demo
Session recording and crypto policies

Red Hat


RED HAT ENTERPRISE LINUX
m1
Session Recording
The intermediate issuer certificate for this site is signed using SHA-1.

</body>
</html>
[root@m1 user]# update-crypto-policies --set FUTURE
Setting system policy to FUTURE
Note: System-wide crypto policies are applied on application start-up.
It is recommended to restart the system for the change of policies
to fully take place.
[root@m1 user]# curl https://sha1-intermediate.badssl.com
curl: (60) SSL certificate problem: EE certificate key too weak
More details here: https://curl.haxx.se/docs/sslcerts.html
curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
[root@m1 user]# exit
[user@m1 ~]$ logout
Connection to m1.cockpit.lan closed.
[admin@m1 ~]$ logout
Connection to m1.cockpit.lan closed.
[stef@dragon demo]$

Red Hat