

# TOP 10 SECURITY CHANGES IN RED HAT ENTERPRISE LINUX 8

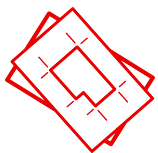
Mark Thacker

Principal Technical Product Manager, Red Hat Enterprise Linux security

# WHAT WE ARE TALKING ABOUT

## Red Hat® Enterprise Linux® 8, of course

---



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 & FIPS validations, NBDE & LUKS disk crypto, AIDE, IMA, Identity Management, Web SSO etc.

---

# AGENDA

- › Compiler flags and static code analysis
- › Consistent and strong crypto policy
- › FIPS mode made easy
- › Smart cards and HSMs
- › TLS 1.3 systemwide
- › Libssh: SSH communications
- › Software identification (SWID) tags
- › Session recording
- › Finer-grained SELinux support
- › Trusted platform module usage

# 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 entire code base
  - Preventing security flaws before shipping and improving the upstream open source
- New compiler flags to prevent stack smashing and mitigate memory corruption
  - Full Address Space Layout Randomization (ASLR)
  - Providing control-flow integrity hardware support
  - Providing full address space layout randomization on all of Red Hat Enterprise Linux via position-independent execution (PIE) and RELRO flags

## Guidance

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

# CONSISTENT AND STRONG CRYPTO POLICY

## 4 policies

- 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

LEGACY

DEFAULT

FIPS 140

FUTURE

# 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 : TLS 1.1

## LEGACY allows TLS 1.1

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

## DEFAULT, FIPS & 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.
```

# TRADITIONAL FIPS MODE ENABLING

Very manual, not easily automated, subject to errors

## 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
```



# 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

# CONSISTENT CONFIGURATION

For smart cards and hardware security modules



## Problems

- How can my systems be hardened against Heartbleed-style attacks?
- How do I set up my smart card or hardware security module (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

# EXAMPLE: SMART CARDS WITH OPENSCH

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%20Token%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

# 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

More coming in future, including Go

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

# LIBSSH: THE LIBRARY FOR SSH COMMUNICATIONS

## Problem

### **Applications need programmatic access to remote systems**

- SSH is the de facto remote access protocol
- Applications need to contact remote systems (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?)



# SOFTWARE ID (SWID) TAGS

## Problem

### **How to perform software inventory management and enforce whitelisting across the enterprise**

- SWID tags provide a means to consistently identify software, its origin, and manufacturer
  - Used by strongSwan, IBM 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 whitelisting for federal governments

# EXAMPLE 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)">
...
```

# 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](https://tagvault.org)

# SESSION RECORDING

## Enabling security compliance and auditing

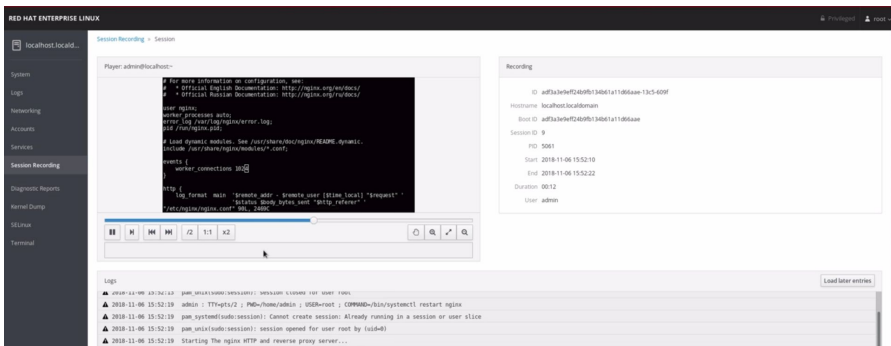
---

- A terminal session recording solution integrated with auditing
- Solving the problem of recording both input and output along with environment and state of system
  - Also preserves text window resizing and timing
- Records events as JSON-formatted audit records via file or syslog
  - Allows records to be quickly, securely exported off system for tamper-proofing
- Selectable on a per-user, per-group basis
  - Integrated with sssd and Identity Management
- Playback via terminal and web console

# SESSION RECORDING EXAMPLE AND NEXT STEPS

## Example

- Playback with actual audit events displayed in-line
- Future integration into centralized logging effort



## 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

# FINE-GRAINED SELINUX CONTROLS

## Problem

### 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
- Watch for even more enhanced Container-specific SELinux changes

# TRUSTED PLATFORM MODULE (TPM) USAGE

## Problem

### 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 / hardware root of trust? Look into upstream community [Keylime.org](https://keylime.org)

# RECAP

- › Compiler flags and static code analysis
- › Consistent and strong crypto policy
- › FIPS mode made easy
- › Smart cards and HSMs
- › TLS 1.3 systemwide
- › Libssh: SSH communications
- › Software identification (SWID) tags
- › Session recording
- › Finer-grained SELinux support
- › Trusted platform module usage



# RESOURCES

- Red Hat Product Security [secalert@redhat.com](mailto:secalert@redhat.com)
- Customer Access Portal <https://access.redhat.com/security/>
- Red Hat Hands-on Security Lab <https://red.ht/securitylabs>
- Red Hat Enterprise Linux 8 <https://redhat.com/rhel/>

# THANK YOU

Mark Thacker

mthacker@redhat.com

@thackman



[linkedin.com/company/red-hat](https://linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://facebook.com/redhatinc)



[twitter.com/RedHat](https://twitter.com/RedHat)

DEMO

# Session Recording and Crypto Policies

RED HAT ENTERPRISE LINUX

System

Log

Networking

Accounts

Services

Session Recording

Diagnostic Tools

Kernel Diagnostics

Tools

Web Services

Remote

```
The intermediate issuer certificate for this site is signed using SHA-1.
</div>
</body>
</html>
[root@ml 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@ml 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@ml user]# exit
[user@ml ~]$ logout

Connection to ml.cockpit.lan closed.
[admin@ml ~]$ logout
Connection to ml.cockpit.lan closed.
[stef@dragon demo]$
```