Thomas Sluijter
29 October 2017

RedHat EX413 cheat sheet

Important things I should remember

Here's the important stuff to take away, grouped by the objectives set out by RedHat.

OBJECTIVE: "*Identify Red Hat Common Vulnerabilities and Exposures (CVEs) and Red Hat Security Advisories (RHSAs) and selectively update systems based on this information*"

Refer to: **man -s8 yum-security**

- **yum updateinfo list**
- **yum update --advisory=$RH-ID**
- **yum update --cve=$CVE**

Applying security-related updates only:

- Install only the security updates: **yum -y update --security**
- Install packages that have errata: **yum -y update-minimal --security**
- The latter updates packages to last security update, which is not necessarily the latest version of the package!

OBJECTIVE: "*Verify package security and validity*"

Validate your installed packages, to find files that have changed: **rpm -Va**

Your YUM repository GPG key files are not used by RPM. RPM has its own, internal GPG key database! In order for Yum to properly check signatures when installing, you need to enforce **gpgcheck=1**, point to your accepted key file and later import the key into RPM.

NOTE FROM REDHAT: "*Package signatures are only checked by yum if the package is downloaded from a repository (which has checking enabled). This happens if the package is specified as a name or name-version-release on the yum command line. If the yum command line names a file or URL instead, or the rpm command is used, no signature check is performed in current versions of Red Hat Enterprise Linux, Fedora, or CentOS.*"

Checking the consistency and validity of an RPM package file:

- **rpm -K $FILE.rpm**, or **rpm --checksig $FILE.rpm** (with or without **-v**)
- When -Kv says "NOKEY", signing the package went wrong.
- When all is well you'll see: file-ver.rpm: (sha1) dsa sha1 md5 gpg OK
- Otherwise, uppercase means there's a problem!

file.rpm: (SHA1) DSA sha1 md5 (GPG) NOT OK (MISSING KEYS: GPG#db42a60e)

PGP keys in your database:

- **rpm -qa gpg-pubkey*** # list current known keys
- **rpm --import $keyfile** # import new key from file
- **rpm --erase $keyID** # erase currently known key
- **rpm -qi gpg-pubkey-$ID** # show verbose info for known key

Reading the contents of a key, from file (not the database):

- **gpg --list-only --import $KEYFILE**

When verifying RPM packages, we need to match either of the group of eight chars of a gpg-pubkey to the last eight chars of the key ID given by the following RPM commands.

Finding PGP keys used to sign RPMs (look for the "Signature" line):

- Already installed package: **rpm -qi $PACKAGE**
- Not yet installed package: **rpm -qpi $FILE.rpm**
- For example:

RSA/SHA512, Tue 27 Jan 2015 11:17:18 PM UTC, Key ID 1054b7a24bd6ec30

Also interesting! Want to find those packages that have not been signed at all?

- **rpm -qa --qf '%{NAME}-%{VERSION}-%{RELEASE} %{SIGPGP:pgpsig} %{SIGGPG:pgpsig}\n' | grep "(none) (none)"**

Inspecting RPM packages before installing them:

- **rpm -qlpv ./$FILE.rpm** # lists package contents
- **rpm -qipv ./$FILE.rpm** # lists package metadata
- **rpm -qp --scripts ./$FILE.rpm** # shows pre- and post-install scripts
- **rpm2cpio ./$FILE.rpm | cpio -idmv** # extracts all package contents

---

OBJECTIVE: "*Identify and employ standards-based practices for configuring file system security, create and use encrypted file systems, tune file system features, and use specific mount options to restrict access to file system volumes.*"

Just remember:

- **cryptsetup [luksFormat, luksOpen, luksClose, luksAddKey]**
- or use **--key-file**
- **/etc/crypttab** = $mapper-label $device $password-or-keyfile

---

OBJECTIVE: "*Configure default permissions for users and use special file permissions, attributes, and access control lists (ACLs) to control access to files*"

Umask can be set in so many different places!

- **umask** command
- **/etc/profile, /etc/bashrc, /etc/init.d/functions, /etc/login.defs**

OBJECTIVE: "*Install and use intrusion detection capabilities in Red Hat Enterprise Linux to monitor critical system files*"

We're talking about AIDE here.

- **/etc/aide.conf**, and don't forget **aide --init**
- It is suggested that you disable prelinking, as it may lead to false positives.
  - "PRELINKING=no" in **/etc/sysconfig/prelink**
  - **/usr/sbin/prelink -ua**

OBJECTIVE: "*Manage user account security and user password security*"

- **/etc/login.defs** applies to new users.
- For existing users, use **chage** and **passwd** (see **--help**).
- To enforce strong password hashing:
  - Add "sha512" to all password lines with **pam_unix.so**.
  - Also edit **/etc/libuser.conf** and add "crypt_style=sha512"
  - Also add "ENCRYPT_METHOD SHA512" to **/etc/login.defs**

OBJECTIVE: "*Manage system login security using pluggable authentication modules (PAM)*"

- Use **man -k pam_** to find the man pages for all modules. Useful stuff to be learned!
- **pam_tally2** is not just a module, also a command line tool.
  - Add it to both the auth and the account lines ...
  - in both **password-auth** and **system-auth**.

OBJECTIVE: "*Configure console security by disabling features that allow systems to be rebooted or powered off using bootloader passwords*"

- **/etc/init/control-alt-delete.override** blocks the use of this key combo.
- **/etc/sysconfig/init** -> change the sushell value to sulogin for root login in SUM.
- **/boot/grub/grub.conf** -> "password --md5 $HASH"

OBJECTIVE: "*Configure system-wide acceptable use notifications*"

- **/etc/issue, /etc/motd**
- You can point SSHd at the issue file with the Banner clause
- Want a warning at Gnome login? Just install gconf-editor!
  - apps -> gdm -> simple-greeter
  - Set banner_message_text and banner_message_enable
  - Right-click on both and "Set as mandatory"

OBJECTIVE: "*Install, configure, and manage identity management services and configure identity management clients*"

Server side:

- Ensure that NTP works!
- **yum install ipa-server bind bind-dyndb-ldap**
- **ipa-server-install [--setup-dns] [--idstart=5000] [--idmax=15000]**
- Leave out **--setup-dns** if the exact domain already has a DNS server.
- On a VM add **--no-ntp** because VMs suck at time keeping.
- Open up firewall ports for LDAP, HTTP(s), NTP, DNS and Kerberos (88, 464)

Client side:

- Ensure that NTP works!
- Point **resolv.conf** at the DNS of the IP box, unless you did not active DNS.
- **yum install ipa-client**
- **ipa-client-install --mkhomedir**

The **ipa** command offers lots of scriptability: **user-add, user-mod, user-find**, etc...

You can update your keytab for example:
ipa-getkeytab -s $SERVER -p $SERVICE/$HOST -k /etc/krb5.keytab

Making sudo work:

- **yum install openldap-clients**
- **kinit admin**
- **ldapsearch -Y GSSAPI -S -h services.ex413.local | grep sysaccounts**
- **ldappasswd -Y GSSAPI -S -h $SERVER \ uid=sudo,cn=sysaccounts,cn=etc,dc=$DOMAIN,dc=$DOMAIN**
- Add "sss" to the sudoers line in **/etc/nsswitch.conf**
- Set the "binddn" and "password" in **/etc/sudoldap.conf**.

To enable automatic home directory creation:

- **/etc/sysconfig/authconfig** -> set "USEMKHOMEDIR=yes"
- This enables PAM modules pam_oddjob_mkhomedir, or pam_mkhomedir.

OBJECTIVE: "*Configure remote system logging services, configure system logging, and manage system log files using mechanisms such as log rotation and compression*"

If you'd like to secure rSyslog with TLS encryption, then setting up the config files and the encryption is a lot of stuff to remember! You're better off installing gnutls, gnutls-tools and rsyslog and then using the documentation in **/usr/share/doc/rsyslog*/**. You will find ready-made examples there!

The filtering options for rsyslog are explained in **/usr/share/doc/rsyslog*/*filter***.

OBJECTIVE: "*Configure system auditing services and review audit reports*"

- It's suggested that you make **/var/log/audit** a separate file system.
- The **man**-page for **auditctl** has great examples.
- **/usr/share/doc/audit*/** also comes to the rescue.
- Basic keystroke logging for root requires PAM:
  - Edit **/etc/pam.d/system-auth** and add
  - **session required pam_tty_audit.so disable=* enable=root**

You can dispatch messages to a remote **auditd** or Syslog:

- **yum install audispd-plugins**
- **/etc/audisp/plugins.d/au-remote.conf** -> set "active = yes"
- **/etc/audisp/audisp-remote.conf** -> set "remote_server" and "port"
- **/etc/audisp/plugins.d/syslog.conf** -> set "active = yes"

More useful commands:

- **ausearch -i -a $IDFIELD**
- **aureport**
- **autrace $COMMAND**
- **sealert -a /var/log/audit/audit.log**

OBJECTIVE: "*Use network scanning tools to identify open network service ports and configure and troubleshoot system firewalling*"

- **nmap**
- **yum install wireshark wireshark-gnome**
- Netcat -> **yum install nc**

Various SELinux things, even if SELinux is not mentioned on the exam objectives.

Adding a different port to SSHd:

- semanage port -a -t ssh_port_t -p tcp 2222
- semanage port -l | grep ssh

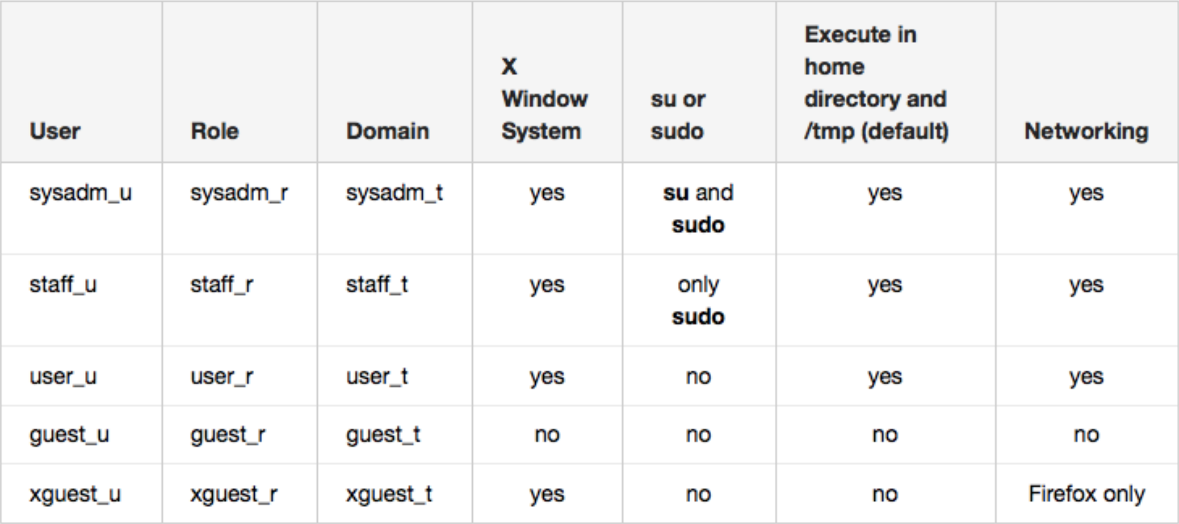
Changing the file context, so Apache can read it:

- semanage fcontext -a httpd_sys_content_t "/web()/.*?"
- restorecon -Rv /web

Booleans:

- semanage boolean -l -> list them all
- setsebool -P $BOOL $VALUE -> change value, Persist
- sesearch -AC | grep ftpd_t -> shows (-C) all allow (-A) permissions involving FTP

The table below shows SELinux User Capabilities, set with **semanage login**. By default, both normal users and root are "unconfined_u".



User	Role	Domain	X Window System	su or sudo	Execute in home directory and /tmp (default)	Networking
sysadm_u	sysadm_r	sysadm_t	yes	**su** and **sudo**	yes	yes
staff_u	staff_r	staff_t	yes	only **sudo**	yes	yes
user_u	user_r	user_t	yes	no	yes	yes
guest_u	guest_r	guest_t	no	no	no	no
xguest_u	xguest_r	xguest_t	yes	no	no	Firefox only