How to Secure RHEL 6.2
Part 1

Motivation

This paper will be a multi-part series on securing Redhat Enterprise Linux 6.2. This idea has been toiling around my head for almost a year and now is the time to get it into print and share with the community. If you have comments or feedback on how I can represent this better, please email me your ideas to the email address listed on the website (securityhardening.com).

Install the Operating System

There is no reason to go into detail about the OS install. Redhat has good documentation on how to install their OS. Below are a couple of screen shots as to how I installed my test system.

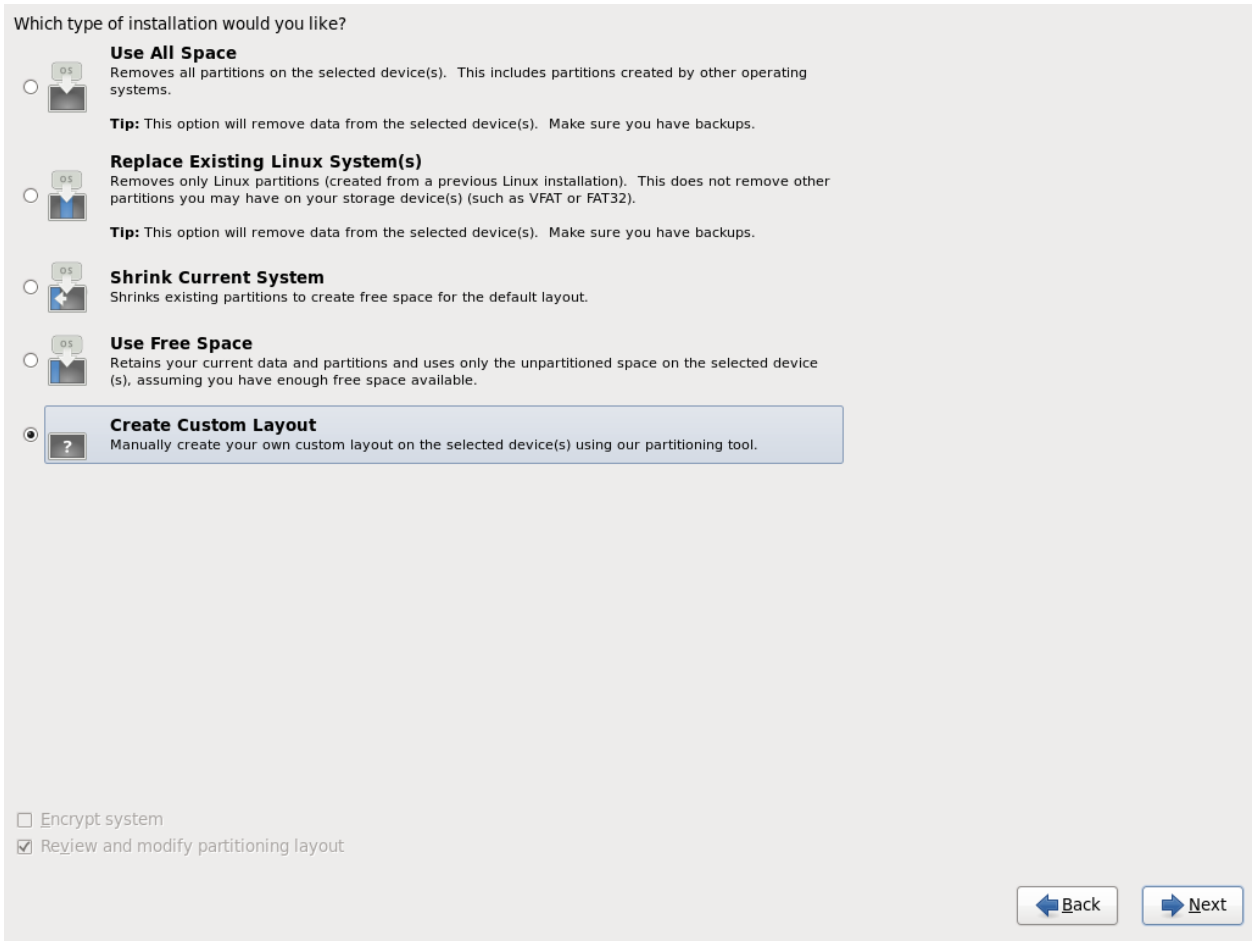

Figure 1

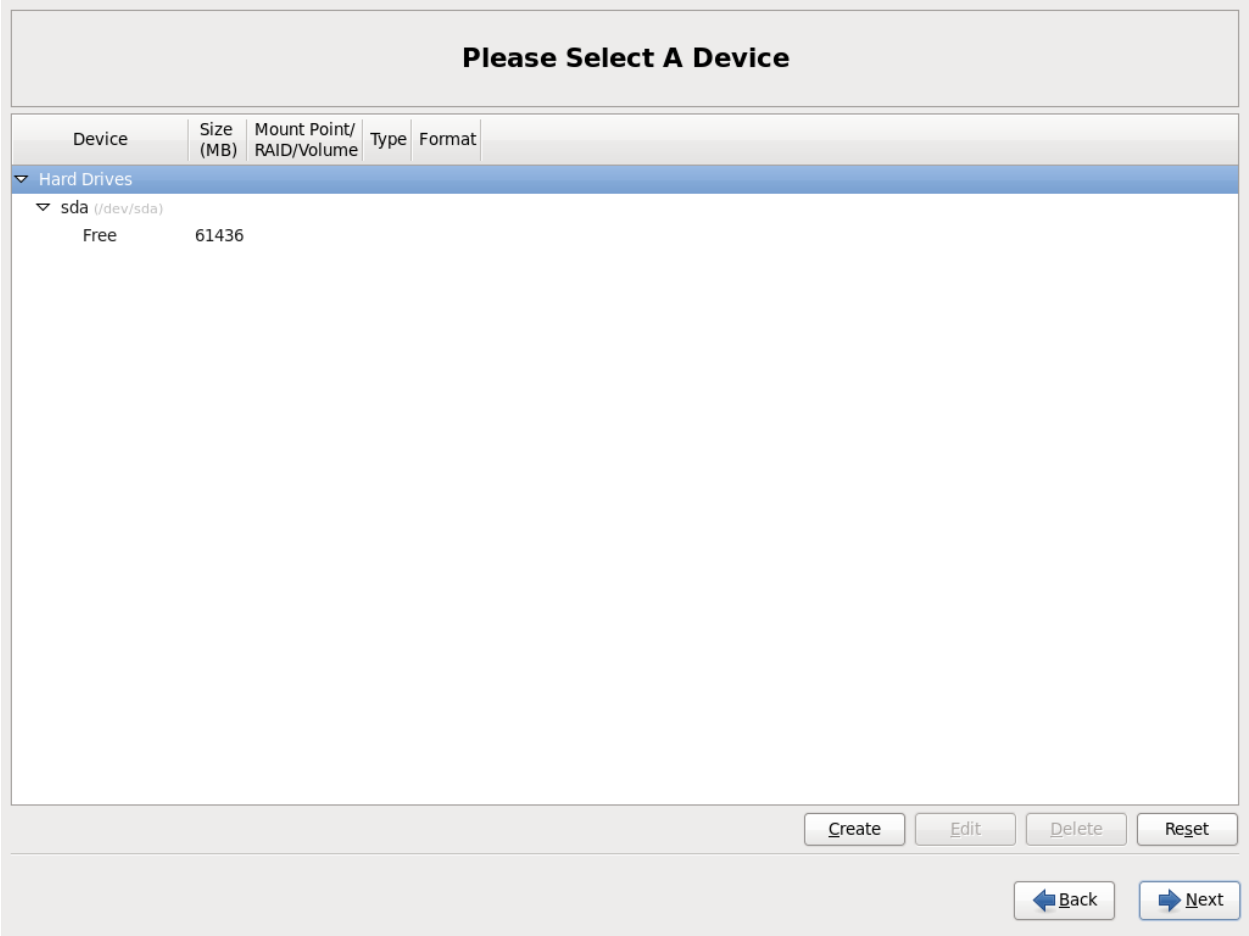

Figure 2

In Figure 1, I always choose the custom layout to set my partitions the way I want them. In Figure 2, this shows a clean hard drive.

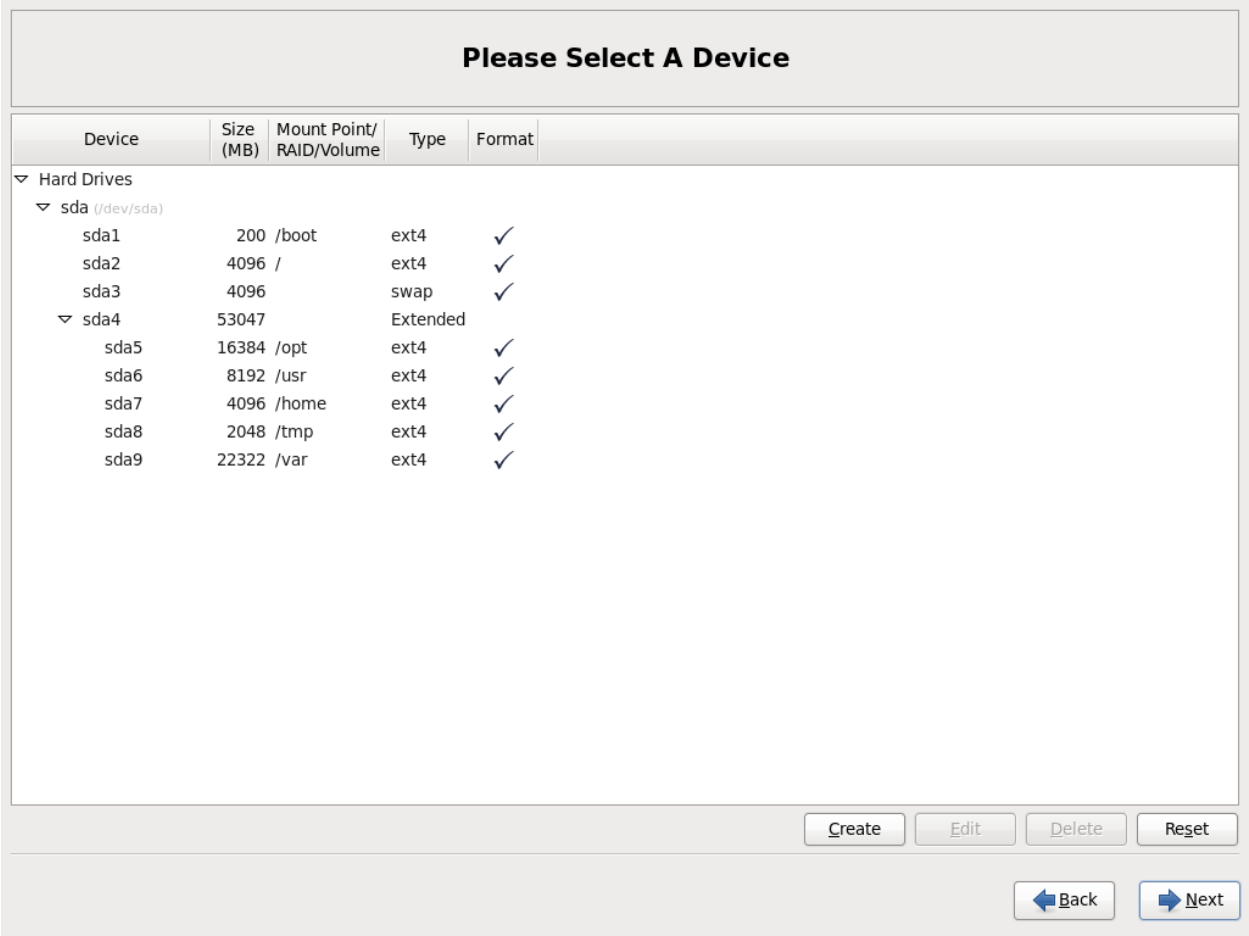

Figure 3

In Figure 3, this represents the layout I chose. Notice that /boot, / and swap are set as primary partitions and /opt, /usr, /home, /tmp and /var are set as extended partitions. The standard I operate under is putting all third party applications under the /opt partition. This to me makes more sense than using the classic /usr/local/, but ultimately the chose comes down to personal preference.

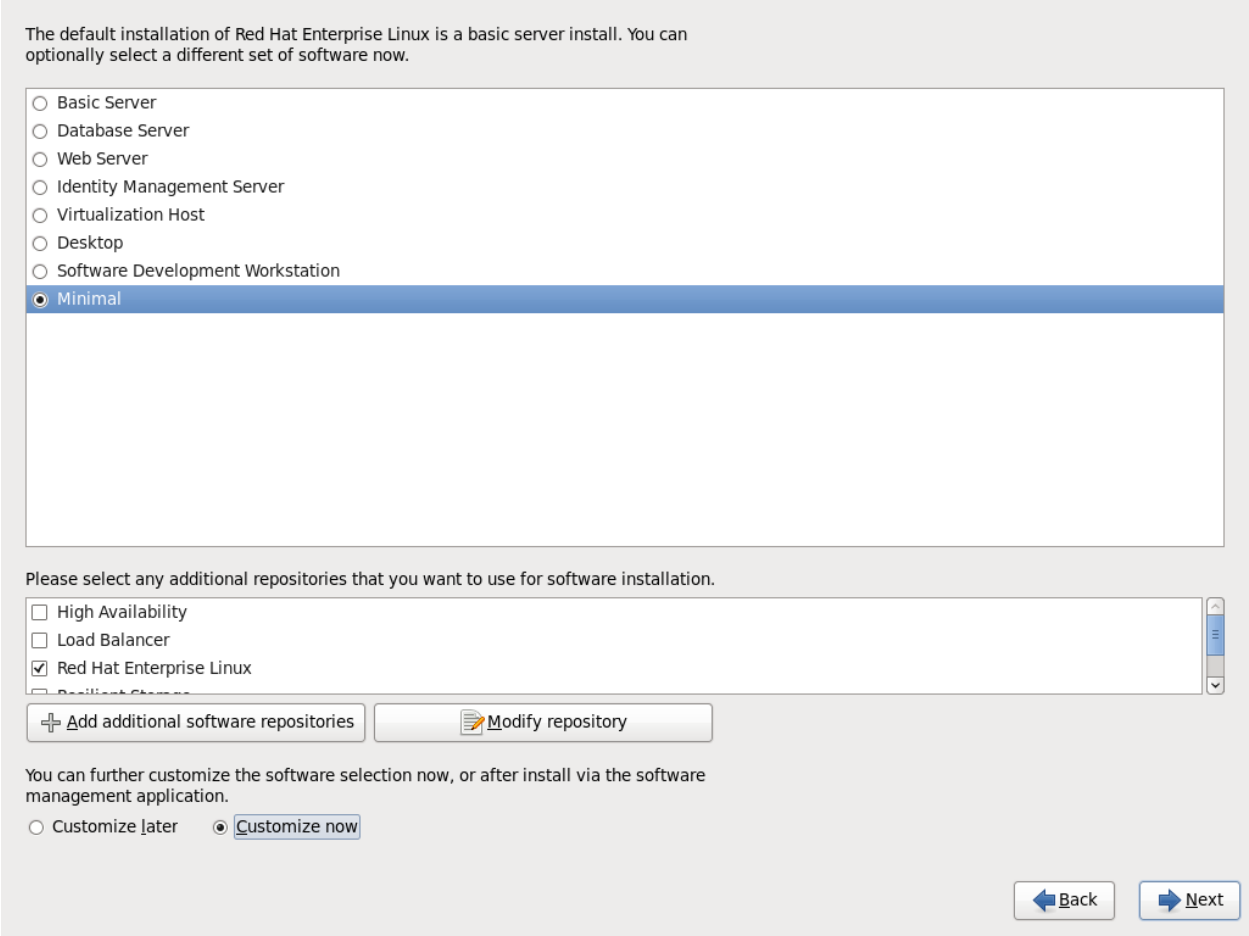

Figure 4

In Figure 4, I always choose the Minimal install to get the minimum amount of packages. This might sound silly, but when it comes to patching, it's a hell of a lot easier to patch a smaller footprint than worry about every package that comes with the default install.

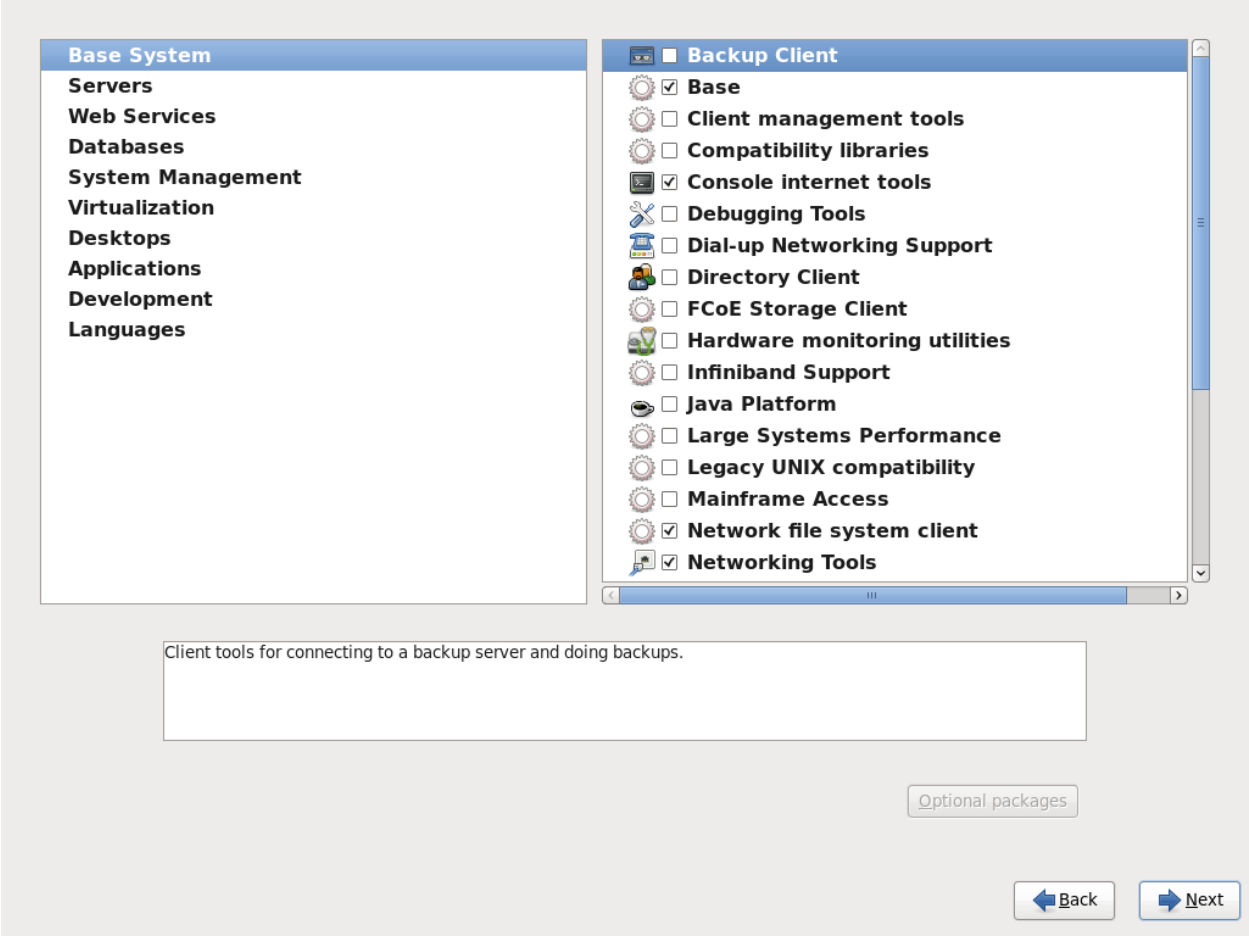

Figure 5

Figure 5 shows a few packages that I chose, again, small is easier to maintain over time.

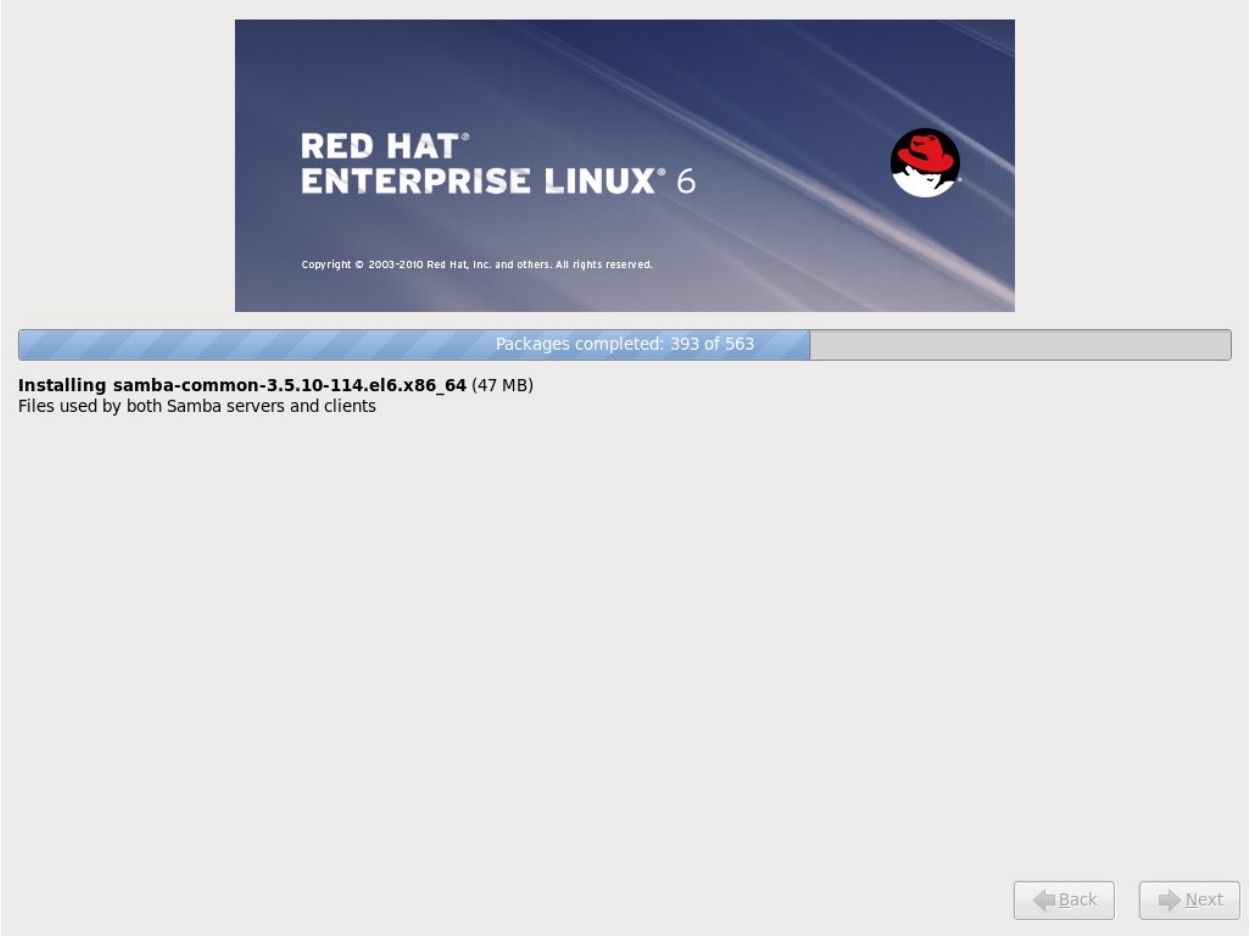

Figure 6

Figure 6 shows the number of packages that the OS is installing.

Secure the Boot Loader

Run the command as the root user, /sbin/grub-md5-crypt.

This will produce the following output.

```
[root@secure62 aide]# /sbin/grub-md5-crypt
Password:
Retype password:
$1$15UNt0$Jq0M2RX49jFmRd.MCH6Xj1
[root@secure62 aide]#
```

Copy the password into your buffer and then open /boot/grub/grub/conf. Add the password line.

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/sda2
#          initrd /initrd-[generic-]version.img
```

```
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
password --md5 $1$15UNt0$Jq0M2RX49jFmRd.MCH6Xj1
hiddenmenu
title Red Hat Enterprise Linux (2.6.32-220.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-220.el6.x86_64 ro root=UUID=55736acd-8506-
47f1-96be-ea0d5d910528 rd_NO_LUKS rd_NO_LVM LANG=en_US.UTF-8 rd_NO_MD quiet
SYSFONT=latarcyrheb-sun16 rhgb crashkernel=auto  KEYBOARDTYPE=pc KEYTABLE=us
rd_NO_DM
        initrd /initramfs-2.6.32-220.el6.x86_64.img
```

Make sure the file permissions and ownership on /boot/grub/grub.conf are set correctly.

```
[root@secure62 aide]# ls -l /boot/grub/grub.conf
-rw------- 1 root root 835 Oct 21 13:46 /boot/grub/grub.conf
[root@secure62 aide]#
```

If you have anything other than this, run:

```
[root@secure62 aide]# chmod 0600 /boot/grub/grub.conf
[root@secure62 aide]# chown root:root /boot/grub/grub.conf
[root@secure62 aide]#
```

A word to the wise, setting the grub password only protects the OS from booting to a different kernel. If a malicious person has physical access to the server, they can boot from a bootable Linux DVD/CD and manipulate the file system any way they choose. Therefore make sure the physical location of the server is secured and depending on the value of the data, only authorized users are capable of booting the system.

Minimize the Boot Sequence

Run the command chkconfig as shown below and minimize unnecessary services.

```
[root@secure62 aide]# chkconfig --list | grep :on
autofs          0:off   1:off   2:off   3:on    4:on    5:on    6:off
cpuspeed        0:off   1:on    2:on    3:on    4:on    5:on    6:off
crond           0:off   1:off   2:on    3:on    4:on    5:on    6:off
haldaemon       0:off   1:off   2:off   3:on    4:on    5:on    6:off
ip6tables       0:off   1:off   2:on    3:on    4:on    5:on    6:off
iptables        0:off   1:off   2:on    3:on    4:on    5:on    6:off
irqbalance      0:off   1:off   2:off   3:on    4:on    5:on    6:off
lvm2-monitor    0:off   1:on    2:on    3:on    4:on    5:on    6:off
mdmonitor       0:off   1:off   2:on    3:on    4:on    5:on    6:off
messagebus      0:off   1:off   2:on    3:on    4:on    5:on    6:off
netfs           0:off   1:off   2:off   3:on    4:on    5:on    6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
ntpd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
postfix         0:off   1:off   2:on    3:on    4:on    5:on    6:off
rsyslog         0:off   1:off   2:on    3:on    4:on    5:on    6:off
sshd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
sysstat         0:off   1:on    2:on    3:on    4:on    5:on    6:off
udev-post       0:off   1:on    2:on    3:on    4:on    5:on    6:off
 [root@secure62 aide]#
```

As an administrator, you should be able to determine what services are necessary for your business use. Playing along, let's say you determine that the cups service is not necessary, run the following command:

```
[root@secure62 aide]# chkconfig --level 2345 cups off
[root@secure62 aide]#
```

Minimize the services down to the smallest amount for the server to run.

Secure the Console and Virtual Terminals

Modify the security file (/etc/securetty) for the console and make it look like this:

```
[root@secure62 etc]# cat securetty
console
vc/1
#vc/2
#vc/3
#vc/4
#vc/5
#vc/6
#vc/7
#vc/8
#vc/9
#vc/10
#vc/11
tty1
#tty2
#tty3
#tty4
#tty5
#tty6
#tty7
#tty8
#tty9
#tty10
#tty11
[root@secure62 etc]#
```

For the Virtual Consoles add the rpm vlock the system and use:

To lock the virtual consoles:

```
[root@secure62 ~]# vlock -c
This TTY is now locked.
Please enter the password to unlock.
root's Password:
```

Or as a user account:

```
[masterf@secure62 ~]$ vlock -c
This TTY is now locked.
Please enter the password to unlock.
masterf's Password:
[masterf@secure62 ~]$
```

To lock all virtual consoles:

```
vlock –a
```

Set a Warning Banner

Run these commands to zero out the following files:

```
[root@secure62 etc]# > /etc/issue
[root@secure62 etc]# > /etc/issue.net
[root@secure62 etc]# > /etc/motd
[root@secure62 etc]#
```

Set /etc/issue to look like this:

```
^[c
\d at \t
Access to this computer system is for authorized personnel only.
Unauthorized use or access to this system is regarded as a
criminal act and is subject to civil and criminal prosecution.
User activities on this system may be monitored without prior notice.
```

To get the "^[c" symbol set correctly, depress "CTRL+v+[" then add the lower case c.

Then set /etc/issue.net to look like this:

```
Access to this computer system is for authorized personnel only.
Unauthorized use or access to this system is regarded as a
criminal act and is subject to civil and criminal prosecution.
User activities on this system may be monitored without prior notice.
```

Groups to remove

Here is the default list of groups on a vanilla install (don't remove all of these):

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
tty:x:5:
disk:x:6:root
lp:x:7:daemon,lp
mem:x:8:
kmem:x:9:
wheel:x:10:root,masterf
mail:x:12:mail,postfix
uucp:x:14:uucp
man:x:15:
games:x:20:
gopher:x:30:
video:x:39:
dip:x:40:
ftp:x:50:
lock:x:54:
audio:x:63:
nobody:x:99:
```

```
users:x:100:
dbus:x:81:
utmp:x:22:
utempter:x:35:
oprofile:x:16:
floppy:x:19:
vcsa:x:69:
rpc:x:32:
abrt:x:173:
cdrom:x:11:
tape:x:33:
dialout:x:18:
haldaemon:x:68:haldaemon
ntp:x:38:
saslauth:x:76:
postdrop:x:90:
postfix:x:89:
rpcuser:x:29:
nfsnobody:x:65534:
stapdev:x:499:
stapusr:x:498:
stap-server:x:155:
arpwatch:x:77:
sshd:x:74:
uuidd:x:497:
tcpdump:x:72:
screen:x:84:
slocate:x:21:
```

Depending on the server's configuration, a lot of these groups are not needed. I'm going to remove:

uucp, gopher, dip, ftp, dialout, rpcuser, nfsnobody, stapdev, stapusr, stap-server

Therefore as root, I can run:

```
for X in uucp gopher dip ftp dialout rpcuser nfsnobody stapdev stapusr stap-server; do groupdel $X; done
```

If you get an error, manually delete the group's line out of /etc/group.

Clean out /etc/passwd and /etc/shadow as necessary. You could delete the users first and then you won't get the error messages from groupdel. Your call as to how you wish to perform this. Another thought that I should share is /etc/passwd has a lot of these accounts set with the shell of /sbin/nologin. If you choose to leave these accounts, make sure they have their shells set to nologin. The benefit of having this set is that if an attacker tries to use these accounts, they will log to /var/log/messages with an error (therefore alerting you that someone is trying to break into your system with that or those accounts). Again, the theme for this work is small is beautiful; hence the focus on removing unneeded accounts and groups.

Securing Passwords

Password complexity and setting rules used to be a pain in the arse. Now days with PAM, this is much less arduous.

Default /etc/pam.d/passwd

```
#%PAM-1.0
auth       include      system-auth
account    include      system-auth
password   substack     system-auth
-password   optional    pam_gnome_keyring.so
```

New /etc/pam.d/passwd

```
#%PAM-1.0
account    required     /lib64/security/pam_unix.so
password   required     /lib64/security/pam_cracklib.so retry=5 →
minlen=15 dcredit=-1 ucredit=-1 ocredit=-1 lcredit=0 difok=3
password    required    /lib64/security/pam_unix.so sha512 shadow →
use_authtok remember=24 nullok
```

Perform a Google search for “pam_cracklib and syntax”. This will help explain the options I used above if needed. Finally .. PAM rocks!

Modify /etc/login.defs and change the following from:

```
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_MIN_LEN    5
PASS_WARN_AGE   7
```

To (meet your organizations policies):

```
PASS_MAX_DAYS   60
PASS_MIN_DAYS   0
PASS_MIN_LEN    15
PASS_WARN_AGE   7
```

Using AIDE to baseline your system

Install the RPM for AIDE from the install DVD or CDs.

Then run, in order:

```
nice -17 aide --init --config=/etc/aide.conf
cp /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

Once the system is set, run a monthly file system integrity check with:

```
nice -17 aide -C --config=/etc/aide.conf
```

The walk away with AIDE is you get free software that performs what tripwire (tripwire.com) used to do in order to monitor your system for file changes. What I like doing is prior to installing third party applications, say something like Oracle Database 11R2 for Linux, I will run AIDE to baseline the system and then after the application is installed, patched and configured, I run AIDE again to report the delta. It’s interesting and solid information to know where all of the files are laying on your file system, especially when you get files in hidden directories that screw up re-installs. Having this knowledge allows you to thoroughly clean out your file system of any crap the vendor conveniently forgot to delete during the uninstall process. Furthermore, the Configuration Management fascists like having the

knowledge of what was modified on the filesystem and AIDE affords the sys admin to quickly generate the documentation necessary to satisfy CM's insatiable greed for more documentation.

In the next paper, I will be covering the software firewall, IPTables that comes default on RHEL 6.2.

If you find any in-accuracies or misrepresentations, please email me so that I can correct my work and keep a better paper online.