Linux Hardening

Securing your CentOS 6 server

November 2012

Remember: OS hardening is only a step in the right direction…

- Network security (VLANs, IPS, firewalls, etc.)
- IT Processes around accesses, Deny all, only allow and use when needed
- Human factor (e.g. e-mailing password)
- In-house software development hardening (SQL injections, etc.)
- COTS configurations (MySQL, Apache, etc.)
- Physical server access (BIOS, GRUB, etc.)
- Providing Internet access from a server

General rule: Only install what is needed

- E.g. Do you really need X-window on a real-time system? (performance and security issues)
- Installing unnecessary software also installs additional security flaws
- CentOS 6.3 minimal:

  `CentOS-6.3-x86_64-minimal.iso`

  (The minimal install iso media is an alternative install to the main CentOS-6.0 distribution and comes with a trimmed down, preselected rpm list. However, it still runs off the standard installer, with all the regular features that one would expect from the main distribution, except the rpm selection screen has been disabled.)

  `http://mirror.centos.org/centos/6/isos/x86_64/`

Installing CentOS 6.3 minimal

- Step 1) Burn ISO and boot
- Step 2) Follow wizard and finish setup
- Step 3) Test server Internet connectivity
- Step 4) “yum update –y” and reboot
- Step 5) OS Hardening
- Step 6) Install/configure your software
- Step 7) Configure firewall

First steps…

- Temporarily stop firewall

```
% service iptables stop
```

- Install some tools

```
% yum install sudo perl ntp crontabs sendmail mlocate wget -y
```

- Install EPEL repository

```
% wget http://fedora.mirror.nexicom.net/epel/6/i386/epel-release-6-7.noarch.rpm
% yum install epel-release-6-7.noarch.rpm -y
% yum repolist
```

```
...
repo id                                repo name                                          status
base                                   CentOS-6 - Base                                    6,346
epel                                   Extra Packages for Enterprise Linux 6 - x86_64     8,029
extras                                 CentOS-6 - Extras                                  6
updates                                CentOS-6 - Updates                                 820
```

Configure users

- Configure password, idle and timeout policies

```
% perl -npe 's/PASS_MAX_DAYS.*/PASS_MAX_DAYS 180/' -i /etc/login.defs
% perl -npe 's/PASS_MIN_LEN.*/PASS_MIN_LEN 8/' -i /etc/login.defs
% perl -npe 's/LOGIN_TIMEOUT.*/LOGIN_TIMEOUT 30/' -i /etc/login.defs
% perl -npe 's/PASS_WARN_AGE.*/PASS_WARN_AGE 7/' -i /etc/login.defs
```

(These defaults still can be overridden by *chage* or *passwd* commands)

```
% vi /etc/profile.d/security.sh
      TMOUT=900
      readonly TMOUT
      export TMOUT
      readonly HISTFILE
% chmod +x /etc/profile.d/security.sh
% vi /etc/ssh/sshd_config
      ClientAliveInterval 900
      ClientAliveCountMax 0
% service sshd restart
```

Configure users (cont'd)

- Manage user accounts

  e.g. our scenario:

  servsupp used by administrators / appssupp used by application maintainers

```
% userdel shutdown
% userdel halt
% userdel games
% userdel operator
% userdel ftp
% userdel gopher
% groupadd support
% useradd servsupp -G support
% useradd appssupp -G support
% passwd servsupp
% passwd appssupp
```

- Make sure no non-root accounts have UID set to 0

```
% awk -F: '($3 == "0") {print}' /etc/passwd
```

Remote access configuration

- Disable unneeded SSHD authentication methods

```
% sed -i 's/.*ChallengeResponseAuthentication.*no/ChallengeResponseAuthentication
yes/g' /etc/ssh/sshd_config
% sed -i 's/.*GSSAPIAuthentication.*yes/GSSAPIAuthentication no/g'
/etc/ssh/sshd_config
```

- Disable direct root login

```
% sed -i 's/.*PermitRootLogin.*yes/PermitRootLogin no/g' /etc/ssh/sshd_config
```

- Only allow permitted users using a valid password

```
% sed -i 's/.*AllowUsers.*/AllowUsers appssupp servsupp/g' /etc/ssh/sshd_config
% sed -i 's/.*PermitEmptyPasswords.*yes/PermitEmptyPasswords no/g'
/etc/ssh/sshd_config
```

- Change SSHD listening port (e.g. 8642)

```
% sed -i 's/[#]\s*Port 22/Port 8642/g' /etc/ssh/sshd_config
% /etc/init.d/sshd restart
```

(then reconnect using « servsupp » on port 8642 and do % su)

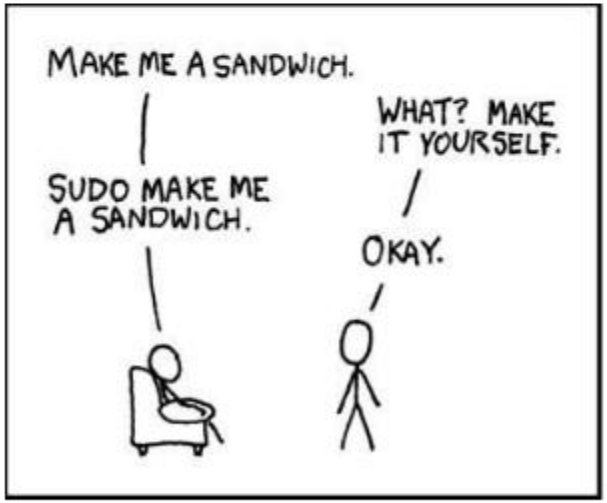
Specify what users can do

- Configure Visudo

```
% visudo
# As an example, append the following line:
%support ALL=NOPASSWD:/etc/init.d/httpd start, /etc/init.d/httpd
stop, /etc/init.d/httpd restart, /sbin/services httpd restart
```

- Test it

```
% su - appssupp
% whoami
% sudo /etc/init.d/httpd restart
```



Credit to: http://xkcd.com/149

Non-SSL software removal

- Remove all non SSL based servers

```
% yum erase xinetd inetd tftp-server ypserv telnet-server rsh-server
```

- A sniffer can easily get your password
- All in and out communications should be encrypted whenever possible
- Use **s**sh, **s**ftp, http**s**, **s**cp, etc.

Services & packages removal

- Verify listening network ports

```
% netstat -tulpn
```

- Verify currently installed packages, e.g.:

```
% yum list installed | more
% yum remove <package>
% yum grouplist
% yum groupremove "X Window System"
```

- Disable unnecessary services

```
% chkconfig --list
% chkconfig <service name> off
% for i in rpcbind restorecond nfslock  lldpad fcoe rpcidmapd; do service $i stop; chkconfig $i off; done
```

- Turn off IPv6 if not needed

```
% vi /etc/sysconfig/network-scripts/ifcfg-eth0
                IPV6INIT=no
                IPV6_AUTOCONF=no
```

Prevent users to schedule tasks

- Prevent anybody but root to run cron or *at* tasks

```
% touch /etc/cron.allow
% chmod 600 /etc/cron.allow
% awk -F: '{print $1}' /etc/passwd | grep -v root > /etc/cron.deny
% touch /etc/at.allow
% chmod 600 /etc/at.allow
% awk -F: '{print $1}' /etc/passwd | grep -v root > /etc/at.deny
```

Narrowing rights

- Narrow down rights for system files and folders

```
% chmod 700 /root
% chmod 700 /var/log/audit
% chmod 740 /etc/rc.d/init.d/iptables
% chmod 740 /sbin/iptables
% chmod -R 700 /etc/skel
% chmod 600 /etc/rsyslog.conf
% chmod 640 /etc/security/access.conf
% chmod 600 /etc/sysctl.conf
```

Verifying file system

- Identify unwanted SUID and SGID Binaries

```
% find / \( -perm -4000 -o -perm -2000 \) -print
% find / -path -prune -o -type f -perm +6000 -ls
```

- Identify world writable files

```
% find /dir -xdev -type d \( -perm -0002 -a ! -perm -1000 \) -print
```

- Identify orphaned files and folders

```
% find /dir -xdev \( -nouser -o -nogroup \) -print
```

Welcome messages

- Make your point on login

```
% cat >/etc/issue << EOF
USE OF THIS COMPUTER SYSTEM, AUTHORIZED OR UNAUTHORIZED, CONSTITUTES CONSENT TO
MONITORING OF THIS SYSTEM. UNAUTHORIZED USE MAY SUBJECT YOU TO CRIMINAL PROSECUTION.
EVIDENCE OF UNAUTHORIZED USE COLLECTED DURING MONITORING MAY BE USED FOR
ADMINISTRATIVE, CRIMINAL, OR OTHER ADVERSE ACTION. USE OF THIS SYSTEM CONSTITUTES
CONSENT TO MONITORING FOR THESE PURPOSES.
EOF
% cp /etc/issue /etc/issue.net
```

- Now configure SSHD

```
% vi /etc/ssh/sshd_config
      Banner /etc/issue.net
% service sshd restart
```

Tune kernel parameters

- **Sysctl** is an interface for examining and dynamically changing parameters in the BSD and Linux operating systems
- Edit sysctl.conf to optimize kernel parameters

  (see http://www.mjmwired.net/kernel/Documentation/networking/ip-sysctl.txt for descriptions)

- **ICMP Redirect**

  This is where an attacker uses either the ICMP "Time exceeded" or "Destination unreachable" messages. Both of these ICMP messages can cause a host to immediately drop a connection. An attacker can make use of this by simply forging one of these ICMP messages, and sending it to one or both of the communicating hosts. Their connection will then be broken. The ICMP "Redirect" message is commonly used by gateways when a host has mistakenly assumed the destination is not on the local network. If an attacker forges an ICMP "Redirect" message, it can cause another host to send packets for certain connections through the attacker's host.

Tune kernel parameters (cont'd)

- ## Synflood

  A Synflood exploits a server by sending a flood of SYN packets, but ignoring the SYN/ACK that come back in response. When this happens, the connection goes into SYN_RCVD state on the server, which is held open until it receives an ACK reply from the Client. If an ACK reply is never received, then it will go into TIME_WAIT state on the server, causing it to be held open even longer. Imagine if a Client sent several hundred (or even thousands) of SYN packets, only to ignore the SYN/ACK reply. Generally, this form of attack targets a particular service to exploit the built in connection limits built into most services. This type of attack will not generally cause a server to go out of memory, but will make the service being targeted, usually Apache, go unresponsive.

  Client [SYN] ----------> Server

  Client <---------- [SYN/ACK] Server

  Client [ACK] ----------> Server

- ## UDP Floods

  UDP attacks are quite effective, as UDP is a state-less protocol. UDP just sends the traffic without the initial overhead of the three way handshake in TCP. In addition to no initial overhead, UDP also has no verification that the packet was received. Normally, when you send a UDP packet with the "DNS Query" bit set to port 53 of a nameserver, it will reply with DNS results if the nameserver is active. However, if you send a UDP packet to a port that doesn't have a service listening on it, the remote host will send back an ICMP Destination Unreachable packet (http://en.wikipedia.org/wiki/ICMP_Destination_Unreachable). It does this for EACH and EVERY UDP packet sent. As floods go, you can imagine sending a whole bunch of UDP packets would generate quite lot of return traffic, now that you don't need state.

Tune kernel parameters (cont'd)

```
cat << 'EOF' >> /etc/sysctl.conf
# drop icmp redirects
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
# double the syn backlog size
net.ipv4.tcp_max_syn_backlog = 2048
# ignore ping broadcasts
net.ipv4.icmp_echo_ignore_broadcasts = 1
# drop the source routing ability
net.ipv4.conf.all.accept_source_route = 0
# log packets destinated to impossible addresses
net.ipv4.conf.all.log_martians = 1
# ignore bogus icmp error responses
net.ipv4.icmp_ignore_bogus_error_responses = 1
```

Tune kernel parameters (cont'd)

```
# drop packets that come in using a bad interface
# (they will be logged as martian)
net.ipv4.conf.all.rp_filter = 1
# don't send timestamps
#net.ipv4.tcp_timestamps = 0
# Kernel threads
kernel.threads-max = 163840
# Socket buffers
net.core.wmem_default = 655360
net.core.wmem_max = 5242880
net.core.rmem_default = 655360
net.core.rmem_max = 5242880
# netdev backlog
net.core.netdev_max_backlog = 4096
# Semafores
kernel.sem="250 32000 32 1024"
# Socket buckets
net.ipv4.tcp_max_tw_buckets = 163840
# Controls source route verification
net.ipv4.conf.all.rp_filter = 1
# Do not accept source route
net.ipv4.conf.all.accept_source_route = 0
# Increase port range
net.ipv4.ip_local_port_range = 2000 65000
EOF
```

Network Time Protocol

- Make sure your server has the right time set at any moment

Network Time Protocol (**NTP**) is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks.

NTP uses Marzullo's algorithm and is designed to resist the effects of variable latency. NTP can usually maintain time to within tens of milliseconds over the public Internet, and can achieve 1 millisecond accuracy in local area networks under ideal conditions.

```
% chkconfig ntpd on
% sed -i 's/server 0.*/server 0\.north\-america\.pool\.ntp\.org/g' /etc/ntp.conf
% sed -i 's/server 1.*/server 0\.north\-america\.pool\.ntp\.org/g' /etc/ntp.conf
% sed -i 's/server 2.*/server 0\.north\-america\.pool\.ntp\.org/g' /etc/ntp.conf
% sed -i 's/server 3.*/server 0\.north\-america\.pool\.ntp\.org/g' /etc/ntp.conf
% /etc/init.d/ntpd start
% ln -sf /usr/share/zoneinfo/America/Montreal /etc/localtime
```

Security policies

Security-Enhanced Linux (**SELinux**) is a Linux feature that provides the mechanism for supporting access control security policies, including United States Department of Defense-style mandatory access controls, through the use of Linux Security Modules (LSM) in the Linux kernel.

- Enabled by default, should you disable it?
- Difficult to configure with some software, transparent with others, google it first
- Disabling SELinux

```
% vi /etc/selinux/config
      "SELINUX=disabled"
```

- Configuration files

See: http://selinuxproject.org/page/ConfigurationFiles

Ban bad IPs

Fail2ban scans log files (e.g. /var/log/apache/error_log) and bans IPs that show the malicious signs -- too many password failures, seeking for exploits, etc. Generally Fail2Ban then used to update firewall rules to reject the IP addresses for a specified amount of time, although any arbitrary other **action** (e.g. sending an email, or ejecting CD-ROM tray) could also be configured. Out of the box Fail2Ban comes with **filters** for various services (apache, curier, ssh, etc).

Ref: `http://www.fail2ban.org`

- Example, regex scanning for Asterisk PBX log files:

```
NOTICE.* .*: Registration from '.*' failed for '<HOST>' - Wrong password
```

- Check current bans:

```
% /usr/bin/fail2ban-client status ssh-iptables
```

Ban bad IPs (cont'd)

- Installation steps

```
% yum install fail2ban
% chkconfig fail2ban on
% vi /etc/fail2ban/jail.conf
        [ssh-iptables]
        enabled  = true
        filter   = sshd
        action   = iptables[name=SSH, port=8642, protocol=tcp]
                   sendmail-
whois[name=SSH, dest=support@yourdomain.com, sender=fail2ban@yourdomain.net]
        logpath  = /var/log/secure
        maxretry = 5
% vi /etc/fail2ban/fail2ban.conf
        logtarget = /var/log/fail2ban.log
% service fail2ban start
```

Log watching & reporting

Logwatch is a customizable log analysis system. Logwatch parses through your system's logs and creates a report analyzing areas that you specify.

Ref: `http://sourceforge.net/projects/logwatch/files/`

- Too much information to consider manual analysis
- Sends daily e-mails to you
- Fail2ban & logwatch

Ref: `http://sourceforge.net/projects/fail2ban/files/logwatch/`

- Even get yesterday's installed packages list

Log watching & reporting (cont'd)

```
% yum install logwatch -y

% vi /usr/share/logwatch/default.conf/logwatch.conf
        MailTo = support@yourdomain.com
        MailFrom = Logwatch@yourdomain.com

% crontab -e
         0 5 * * * /usr/sbin/logwatch
```

Firewall setup

iptables is a user space application program that allows a system administrator to configure the tables provided by the Linux kernel firewall (implemented as different Netfilter modules) and the chains and rules it stores. Different kernel modules and programs are currently used for different protocols; *iptables* applies to IPv4, *ip6tables* to IPv6, *arptables* to ARP, and *ebtables* to Ethernet frames

- Create a check list for your applications
- Create a shell script with all of your rules
- Start *iptables* service and run the script

```
% service iptables start
% ./myrules.sh
```

- Check applications connectivity
- Also configure your hardware firewall!

Firewall (cont'd)

```
cat > myrules.sh << EOF
    #!/bin/bash
    # Flush all current rules from iptables
    iptables -F

    # SSH (8642) - Don't lock ourselves out + Limit of 3 attempts per minute
    iptables -A INPUT -p tcp --dport 8642 --syn -m limit --limit 1/m --limit-burst 3 -j ACCEPT
    iptables -A INPUT -p tcp --dport 8642 --syn -j DROP

    # Set default policies for INPUT, FORWARD and OUTPUT chains
    iptables -P INPUT DROP
    iptables -P FORWARD DROP
    iptables -P OUTPUT ACCEPT

    # Set access for localhost
    iptables -A INPUT -i lo -j ACCEPT

    # Accept packets belonging to established and related connections
    iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

    # ICMP (PING) - Ping flood protection 1 per second
    iptables -A INPUT -p icmp -m limit --limit 5/s --limit-burst 5 -j ACCEPT
    iptables -A INPUT -p icmp -j DROP

    # MySQL (3306)
    iptables -A INPUT -p tcp --dport 3306 -j ACCEPT
    iptables -A INPUT -p udp --dport 3306 -j ACCEPT

    # Save settings
    /sbin/service iptables save

    # List rules
    iptables -L -v
EOF
```

Miscellaneous

- Change the nb of available gettys as there too many available by default

```
% sed -i 's/1\-6/1/g' /etc/sysconfig/init
% sed -i 's/1\-6/1/g' /etc/init/start-ttys.conf
```

- Require root's password for single user mode

```
% echo "~~:S:wait:/sbin/sulogin" >> /etc/inittab
```

- Disable USB mass storage, if you're not using it in your environment

```
% echo "blacklist usb-storage" > /etc/modprobe.d/blacklist-usbstorage
```

- Once a server is up and running, root shouldn't be logging in directly except in emergency situations. These usually require hands at the console, so that's the only place root should be allowed to log in

```
% echo "tty1" > /etc/securetty
```

- Update the system to use sha512 instead of md5 for password protection

```
% authconfig --passalgo=sha512 --update
```

Going forward: security updates

- Regularly install the security patches
- Set your Outlook reminders
- Install *yum* plugin

```
% yum -y install yum-plugin-security
```

- Check and install security updates

```
% yum updateinfo summary
% yum --security check-update
% yum --security update
```

- CentOS 'announce' mailing list

```
http://lists.centos.org/mailman/listinfo/centos-announce
```

Server proximity security

- To consider:

1. The only folks allowed near the server should be directly responsible for it.
2. Don't allow the system to boot from removable media as the default option
3. Require a bios password to change boot options. OS security doesn't matter much if your attacker brings their own OS to the party.
4. Use a password for grub. All the security in the world is for nothing if someone can simply pass some arguments to your loader and disable your security.
5. Require a password for single user mode. Same reason as above.
6. Most servers don't need usb storage devices. Disable the usb-storage driver if possible.

- The end -

support@myowntelco.net
www.myowntelco.net