How to Secure CENTOS 7.1
Part 5

Motivation

This paper is part of a multi-part series on securing CentOS 7.1. This paper focuses on the setup and configuration of rsyslog with the mysql database. This combination is extremely powerful because it allows you the flexibility of connecting a web server to display your audit logs or having another service connect to the database with the purpose of searching the records, i.e. for an entire enterprise worth of servers.

rsyslog

Installation

Use YUM to install the following packages.

```
# yum install -y pcre pcre-devel mysql-server mysql-devel gnutls gnutls-devel
gnutls-utils net-snmp net-snmp-devel net-snmp-libs net-snmp-perl net-snmp-
utils libnet libnet-devel
```

Make a temporary directory that will hold your software:

```
# mkdir /var/tmp/Software
```

Download the software to this new directory:

```
# cd /var/tmp/Software/
# wget http://sourceforge.net/projects/libestr/files/libestr-0.1.0.tar.gz/download
# wget http://www.libee.org/files/download/libee-0.1.0.tar.gz
# wget http://pkgs.fedoraproject.org/repo/pkgs/librelp/librelp-
1.0.0.tar.gz/0b8820f8da639a00c75f5cc8f5d21919/librelp-1.0.0.tar.gz
```

Extract the compressed tarballs:

```
# for X in $(ls *.gz); do tar xzvf $X; done
```

Compile and Install each of the tarballs:

```
# cd libestr-0.1.0
# ./configure --libdir=/usr/lib64/ --includedir=/usr/include --prefix=/usr
# make && make install
```

Next:

```
# cd ../libee-0.1.0
# ./configure --libdir=/usr/lib64/ --includedir=/usr/include --prefix=/usr
# make && make install
```

Finally:

```
# cd ../librelp-1.0.0
# ./configure --libdir=/usr/lib64/ --includedir=/usr/include --prefix=/usr
# make && make install
```

Download EPEL and install:

```
# wget http://dl.fedoraproject.org/pub/epel/5/x86_64/epel-release-5-
4.noarch.rpm
# rpm -ivh epel-release-5-4.noarch.rpm
# yum install libnet.x86_64 libnet-devel.x86_64
```

Now download rsyslog, compile and install rsyslog:

```
# cd /var/tmp/Software
# wget http://www.rsyslog.com/files/download/rsyslog/rsyslog-5.7.9.tar.gz
# tar xzvf rsyslog-5.7.9.tar.gz
# cd rsyslog-5.7.9
# ./configure --enable-regexp --enable-zlib --enable-pthreads --enable-klog \
              --enable-inet --enable-unlimited-select --enable-debug \
              --enable-rtinst --enable-memcheck --enable-diagtools \
              --enable-mysql --enable-snmp --enable-gnutls \
              --enable-rsyslogrt --enable-rsyslogd --enable-extended-tests \
              --enable-mail --enable-imptcp --enable-omruleset \
              --enable-valgrind --enable-imdiag --enable-relp \
              --enable-testbench --enable-imfile --enable-omstdout \
              --enable-omdbalerting --enable-omuxsock --enable-imtemplate \
              --enable-omtemplate --enable-pmlastmsg --enable-omudpspoof \
              --enable-omprog --enable-impstats
# make && make install
```

Configuration

Copy the following content into /etc/init.d/rsyslog:

```
#!/bin/bash
#
# rsyslog        Starts rsyslogd.
# chkconfig: - 12 88
# Provides: $syslog
# Required-Start: $local_fs $network $remote_fs
# Required-Stop: $local_fs $network $remote_fs
# Default-Stop: 0 1 2 3 4 5 6

# Source function library.
. /etc/init.d/functions

RETVAL=0

start() {
        [ -x /usr/local/sbin/rsyslogd ] || exit 5
        #[ -x /usr/local/sbin/rklogd ] || exit 5

        # Do not start rsyslog when sysklogd is running
        if [ -e /var/run/syslogd.pid ] ; then
                echo $"Shut down sysklogd before you run rsyslog";
                exit 1;
        fi

        # Source config
```

```
        if [ -f /etc/sysconfig/rsyslog ] ; then
                . /etc/sysconfig/rsyslog
        else
                SYSLOGD_OPTIONS="-c5"
        fi

        if [ -z "$SYSLOG_UMASK" ] ; then
              SYSLOG_UMASK=077;
        fi
        umask $SYSLOG_UMASK

        echo -n $"Starting system logger: "
        daemon /usr/local/sbin/rsyslogd $SYSLOGD_OPTIONS
        RETVAL=$?
        echo
        #echo -n $"Starting kernel logger: "
        #daemon rklogd $KLOGD_OPTIONS
        #echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/rsyslog
        return $RETVAL
}
stop() {
        #echo -n $"Shutting down kernel logger: "
        #killproc rklogd
        #echo
        echo -n $"Shutting down system logger: "
        killproc rsyslogd
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/rsyslog
        return $RETVAL
}
reload()  {
    RETVAL=1
    syslog=`cat /var/run/rsyslogd.pid 2>/dev/null`
    echo -n "Reloading system logger..."
    if [ -n "${syslog}" ] && [ -e /proc/"${syslog}" ]; then
        kill -HUP "$syslog";
        RETVAL=$?
    fi
    if [ $RETVAL -ne 0 ]; then
        failure
    else
        success
    fi
    echo
    RETVAL=1
    #echo -n "Reloading kernel logger..."
    #klog=`cat /var/run/rklogd.pid 2>/dev/null`
    #if [ -n "${klog}" ] && [ -e /proc/"${klog}" ]; then
        #kill -USR2 "$klog";
    #    RETVAL=$?
    #fi
    #if [ $RETVAL -ne 0 ]; then
        #failure
    #else
        #success
```

```
    #fi
    #echo
    return $RETVAL
}
rhstatus() {
        status rsyslogd
        #status rklogd
}
restart() {
        stop
        start
}

case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart)
        restart
        ;;
  reload|force-reload)
        reload
        ;;
  status)
        rhstatus
        ;;
  condrestart)
        [ -f /var/lock/subsys/rsyslog ] && restart || :
        ;;
  *)
        echo $"Usage: $0 {start|stop|restart|reload|force-
reload|condrestart}"
        exit 2
esac

exit $?
```

Copy the following content into /etc/ rsyslog.conf:

```
# Input Modules
#
$ModLoad impstats.so
$PStatsInterval 300
syslog.info  /var/log/rsyslog/rsyslog-stats
#
$ModLoad immark.so      # provides --MARK-- message capability
$ModLoad imuxsock.so    # provides support for local system logging (via logger command)
$ModLoad imklog.so      # provides kernel logging support (previously done by rklogd)
#
$ModLoad imudp.so       # provides UDP syslog reception
$UDPServerAddress *     # all local interfaces
$UDPServerRun 514       # start UDP server (log server receiver)
#
$ModLoad imtcp.so       # provides TCP syslog reception and GSS-API (if compiled)
$InputTCPServerRun 514  # start TCP server (log server receiver)
#
```

```
$ModLoad imrelp.so      # RELP input
$InputRELPServerRun 20514 # start RELP Protocol
#
$ModLoad imfile.so      # Text file input
$InputFileName /var/log/i-am-a-text-file.log
$InputFileTag my-text-file:
$InputFileStateFile stat-file1
$InputFileSeverity error
$InputFileFacility local7
$InputFilePollInterval 10 # check for new lines every 10 seconds
$InputRunFileMonitor
#
#$ModLoad imgssapi.so   # Plain TCP and GSSAPI
#$ModLoad im1395.so     # Messages via RFC1395

# Output Modules
#
$ModLoad omsnmp.so      # Send SNMP traps
#$actionsnmptransport udp
#$actionsnmptarget 192.168.x.x
#$actionsnmptargetport 162
#$actionsnmpversion 1
#$actionsnmpcommunity public
#*.* :omsnmp:
#
$ModLoad ommysql.so     # Log to MySQL
#
$ModLoad ommail.so      # Send mail
#$ActionMailSMTPServer mail.example.net
#$ActionMailFrom rsyslog@example.net
#$ActionMailTo operator@example.net
#$ActionMailTo admin@example.net
#$template mailSubject,"disk problem on %hostname%"
#$template mailBody,"RSYSLOG Alert\r\nmsg='%msg%'"
#$ActionMailSubject mailSubject
#$ActionExecOnlyOnceEveryInterval 21600
#if $msg contains 'hard disk fatal failure' then :ommail:;mailBody
#
$ModLoad omrelp.so      # Send to another host via RELP
#$ModLoad omlibdbi.so   # Log via generic DB output
#$ModLoad omgss.so      # GSS enabled output

# Globals
$umask 0000
$DirCreateMode 0640
$FileCreateMode 0640
$RepeatedMsgReduction on

$WorkDirectory /var/log/rsyslog  # default location for work (spool) files
$ActionQueueType LinkedList      # use asynchronous processing
$ActionQueueFileName queue       # set file name, also enables disk mode
$ActionResumeRetryCount -1       # infinite retries on insert failure
$ActionQueueSaveOnShutdown on    # save in-memory data if rsyslog shuts down
$MainMsgQueueMaxFileSize 100M
$ActionQueueMaxFileSize 5M

#
# Below find some samples of what a template can do. Have a good
# time finding out what they do [or just tun them] ;)

# A template that resambles traditional syslogd file output:
$template TraditionalFormat,"%timegenerated% %HOSTNAME% %syslogtag%%msg:::drop-last-lf%\n"

# a template useful for debugging format issues
$template DEBUG,"Debug line with all properties:\nFROMHOST: '%FROMHOST%', HOSTNAME: '%HOSTNAME%',
PRI: %PRI%,\nsyslogtag '%syslogtag%', programname: '%programname%', APP-NAME: '%APP-NAME%',
PROCID: '%PROCID%', MSGID: '%MSGID%',\nTIMESTAMP: '%TIMESTAMP%', STRUCTURED-DATA: '%STRUCTURED-
DATA%',\nmsg: '%msg%'\nescaped msg: '%msg:::drop-cc%'\nrawmsg: '%rawmsg%'\n\n"

# A template that resembles RFC 3164 on-the-wire format:
# (yes, there is NO space betwen syslogtag and msg! that's important!)
```

```
$template RFC3164fmt,"<%PRI%>%TIMESTAMP% %HOSTNAME% %syslogtag%%msg%"

# a template resembling traditional wallmessage format:
$template wallmsg,"\r\n\7Message from syslogd@%HOSTNAME% at %timegenerated% ...\r\n
%syslogtag%%msg%\n\r"

# The template below emulates winsyslog format, but we need to check the time
# stamps used. for now, it is good enough ;) This format works best with
# other members of the MonitorWare product family. It is also a good sample
# where you can see the property replacer in action.
$template WinSyslogFmt,"%HOSTNAME%,%timegenerated:1:10:date-rfc3339%,%timegenerated:12:19:date-
rfc3339%,%timegenerated:1:10:date-rfc3339%,%timegenerated:12:19:date-
rfc3339%,%syslogfacility%,%syslogpriority%,%syslogtag%%msg%\n"

# A template used for database writing (notice it *is* an actual
# sql-statement):
$template dbFormat,"insert into SystemEvents (Message, Facility,FromHost, Priority,
DeviceReportedTime, ReceivedAt, InfoUnitID, SysLogTag) values ('%msg%', %syslogfacility%,
'%HOSTNAME%',%syslogpriority%, '%timereported:::date-mysql%', '%timegenerated:::date-mysql%',
%iut%, '%syslogtag%')",sql

$template FileFormat,"%TIMESTAMP:::date-rfc3339% %HOSTNAME% %syslogtag%%msg:::sp-if-no-1st-
sp%%msg:::drop-last-lf%\n"

$template ForwardFormat,"<%PRI%>%TIMESTAMP:::date-rfc3339% %HOSTNAME% %syslogtag:1:32%%msg:::sp-
if-no-1st-sp%%msg%"

# Selector lines are somewhat different from stock syslogd. With
# rsyslog, you can add a semicolon ";" after the target and then
# the template name. That will assign this template to the respective
# action. If no template name is given, a hardcoded template is used.
# If a template name is given, but the template was not defined, the
# selector line is DEACTIVATED.
#----------------------------------------------------------------------

#
# Forward via TCP with maximum compression:
#$AllowedSender TCP, 127.0.0.1, 192.0.2.0/24, [::1]/128, *.example.net, somehost.example.com
#*.*       @@(z9)192.168.x.x:514
# Forward via UDP with maximum compression:
#$AllowedSender UDP, 127.0.0.1, 192.0.2.0/24, [::1]/128, *.example.net, somehost.example.com
#*.*       @(z9)192.168.x.x:514
# Forward via RELP Protocol :
#*.*      :omrelp:192.168.2.4:20514;TraditionalFormat
# Store all log files in MySQL DB  :
*.*       :ommysql:127.0.0.1,Syslog,rsyslog,some-secret-Pa$$w0rd-Not
#


#
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console;TraditionalFileFormat

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog

# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 *

# Save news errors of level crit and higher in a special file.
```

```
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log

#
$IncludeConfig /etc/rsyslog.d/*.conf

#
#if message contains 'network error' then run the restart-network.sh shell script!!!
#:msg, contains, "network error" ^/root/restart-network.sh
```

Activate the services to start on boot:

```
# chkconfig --level 2345 mysqld on
# chkconfig --level 2345 rsyslog on
```

Prepping the MySQL database:

```
# /var/tmp/Software/rsyslog-5.7.9
# mysql -u root -p < plugins/ommysql/createDB.sql
# mysql -u root -p mysql
# GRANT ALL ON Syslog.* TO rsyslog@localhost IDENTIFIED BY 'some-secret-
Pa$$w0rd-Not';
# flush privileges;
```

Log Rotation

Copy the following into /etc/logrotate.d/rsyslog:

```
/var/log/boot.log
/var/log/cron
/var/log/maillog
/var/log/messages
/var/log/secure
/var/log/spooler
{
    sharedscripts
    postrotate
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null
|| true
    endscript
}
```

Restart the services:

```
# systemctl restart mysqld
# systemctl restart rsyslog
```

Test your configuration:

```
# logger "test message to rsyslog"
```

Now run:

```
# tail -10 /var/log/messages
```

You should see:

```
2012-11-23T12:40:37-05:00 rhel57 masterf: test message to rsyslog
```

Now test the database:

```
# mysql -u root -p mysql
# use Syslog
# select * from SystemEvents;
```

You should see something similar to:

```
                                 |       NULL |       NULL | NULL         | NULL
|          NULL |    NULL | NULL             |          NULL |      NULL |
NULL |     NULL |           1 | crond[3280]:             | NULL          | NULL
|     NULL |
| 770 |       NULL | 2012-11-23 12:40:37 | 2012-11-23 12:40:37 |         1 |
5 | rhel57   |  test message to rsyslog
|       NULL |       NULL | NULL         | NULL      |           NULL |    NULL
| NULL             |          NULL |      NULL |     NULL |     NULL |
1 | masterf:                 | NULL          | NULL             |     NULL |
```

Conclusion

There is a lot going on here to get all of the dependencies established, the correct software downloaded, compiled and installed in the right location. The possibilities are endless on how you use this system. With this configuration, it monitors the local system and puts those records into the database. Extending this system to an enterprise solution is not hard, just put in the domain or subnet to monitor and configure the clients to push their syslog to your new rsyslog server. I highly recommend reading this link: http://www.rsyslog.com/doc/manual.html.