

# Check Point automatic MDS backup script with upload to SSH

Vložil/a [cm3l1k1](#) [1], 29 Červenec, 2010 - 15:42

- [Check Point](#) [2]
- [GNU/Linux a BSD](#) [3]
- [Networks & Protocols](#) [4]
- [Programming](#) [5]

```
#!/bin/bash
#
# Check Point automatic MDS backup script with upload to SSH(SCP)/FTP server
# Author: Martin Cmelik (cm3l1k1) 11.1.2010
# Website: www.security-portal.cz (use translator module if needed)
# License: GNU General Public License version 3
#

# THE SCRIPT IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
# MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
# LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
# WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
#
#
# 1. Setup script variables & environment
# 2. Create temporary directory
# 3. mdsstop & backup & mdsstart
# 4. Create backup file SHA1 hash and export file SHA1 hash values
# 5. Transfer backup + sha1 hash files to SCP server
# 6. Change backup files permissions
# 7. Moving backup files to local archive location
# 8. Trim backup folder to last 30 days only
#
# Default directories:
# /var/mdsbackups
# /var/mdsbackups/archives
# /var/mdsbackups/log
# /var/mdsbackups/scripts (but MDS backup script can be anywhere)
# and SSH access without password (SSH keys) to SCP server
#
# If you don't need to backup log files and db_versions, add these two lines
# to $MDSDIR/conf/mds_exclude.dat
# log/*
# db_versions/*
#
# save the script as /var/mdsbackups/scripts/mds_backup_script.sh and schedule
# start in crontab as below to run at 1am every sunday, stderr to stdout
# - add this line to CRONTAB (crontab -e)
# 1 1 * * 0 /var/mdsbackups/scripts/mds_backup_script.sh >
/var/mdsbackups/log/mds_backup_script.log 2>&1
```

## Check Point automatic MDS backup script with upload to SSH

Publikováno na serveru Security-Portal.cz (<https://www.security-portal.cz>)

---

```
#
# Initializing log file
#
echo "---//### Check Point automatic MDS BACKUP script  ###\\---"
echo "---//### BEGIN logfile of last $0 script run  ###\\---"
echo `bin/date`

#
# 1. Setup script variables, exit function & check environment
#
PATH=/usr/local/bin:/usr/bin:/bin
ARCHIVE_DIR=/var/mdsbackups/archives/
BACKUP_DIR=/var/mdsbackups/
# at least 2GB free space in backup_dir
FREESPACE=2048000
HOSTNAME=`hostname`
LOG_DIR=/var/mdsbackups/log/
LOG_FILE=$LOG_DIR"mds_backup_script.log"
LOG_MAIL=XXX_YOUR_MAIL_ADDRESS
SCP_PATH=XXX_SCP_DIRECTORY_PATH
SCP_SERVER=XXX_SCP_SERVER
SCP_USERNAME=XXX_SCP_USERNAME
SMTP_SERVER=XXX_SMTP_SERVER
TEMPDIR="$BACKUP_DIR$(basename $0).$RANDOM.temp/"

#
# Source the Check Point profile for library and paths settings
#
export `grep "CPDIR_PATH=" /etc/init.d/firewall1`
[ -f $CPDIR_PATH/tmp/.CPprofile.sh ] || {
    echo "--- Fatal error: cant find CPprofile.sh !!"
    # We are unable to setup essential variables
    `find / -type f -name sendmail` "MDS backup FAILED on $HOSTNAME, please check!"
-t $SMTP_SERVER -f $HOSTNAME $LOG_MAIL < $LOG_FILE
    exit 2
}
source $CPDIR_PATH/tmp/.CPprofile.sh

#
# now we can find sendmail executable
#
SENDMAIL=`which sendmail`
SENDERERRORLOG="$SENDMAIL -s \"MDS backup FAILED on $HOSTNAME, please check!\" -t
$SMTP_SERVER -f $HOSTNAME $LOG_MAIL < $LOG_FILE"
SENDLOG="$SENDMAIL -s \"MDS backup log from $HOSTNAME\" -t $SMTP_SERVER -f $HOSTNAME
$LOG_MAIL < $LOG_FILE"

#
# Setup MDS environment
#
$MDSDIR/scripts/MDSprofile.sh

#
# End script in case of error and send log file
#
# trap also this exit signals: 1/HUP, 2/INT, 3/QUIT, 15/TERM, ERR
trap exit_on_error 1 2 3 15 ERR

function exit_on_error() {
```

## Check Point automatic MDS backup script with upload to SSH

Publikováno na serveru Security-Portal.cz (<https://www.security-portal.cz>)

---

```
    local exit_status=${1:-$?}
    echo "--- Error: Exiting $0 with $exit_status"
    $SENDERRORLOG
    exit $exit_status
}

#
# Check that needed directories exists
#
for CHECK_DIR in $BACKUP_DIR $ARCHIVE_DIR $LOG_DIR; do
[ -d $CHECK_DIR ] || {
echo "--- Error: directory $CHECK_DIR does not exist! I will create it..."
mkdir -p $CHECK_DIR
}
done

#
# Check enough free space on device
#
df -k $BACKUP_DIR | grep -vi filesystem | awk '{ print $4 }' | while read
ACTUALFREESPACE;
do
[ $ACTUALFREESPACE -gt $FREESPACE ] || {
    echo "--- Error: Not enough free space in backup directory $BACKUP_DIR !!"
    $SENDERRORLOG
    exit 2
}
done

#
# 2. Create temporary directory
#
mkdir $TMPDIR
echo `date +%H:%M` "---### Temporary dir $TMPDIR created ###---"

#
# Changing context, we are now working in TMPDIR!
#
cd $TMPDIR

#
# 3. mdsstop & mds_backup & mdsstart, check exit status of mds_backup
#
echo `date +%H:%M` "---### MDS service is going offline ###---"
$MDSDIR/scripts/mdsstop &&
echo `date +%H:%M` "---### MDS backup in progress... ###---"
# stdout to /dev/null (generates thousands lines)
$MDSDIR/scripts/mds_backup -b > /dev/null &&
echo `date +%H:%M` "---### MDS backup is done, starting MDS services ###---"
$MDSDIR/scripts/mdsstart &&

#
# set BACKUP_FILE variable
#
BACKUP_FILE=`ls $TMPDIR | grep mdsbk.tgz`

#
```

## Check Point automatic MDS backup script with upload to SSH

Publikováno na serveru Security-Portal.cz (<https://www.security-portal.cz>)

---

```
# 4. Create backup file SHA1 hash and export file SHA1 hash values.
#

#
#set SHA1SUM & BACKUP_SHA1_HASH variable
#
SHA1SUM=`shasum $BACKUP_FILE | awk '{ print $1; }'`
BACKUP_SHA1_HASH=$BACKUP_FILE.sha

logger "MDS BACKUP: Backup file $BACKUP_FILE created with shasum $SHA1SUM"
echo $SHA1SUM > $BACKUP_SHA1_HASH
echo $HOSTNAME >> $BACKUP_SHA1_HASH
echo `date +%H:%M` "----### BACKUP: $BACKUP_FILE created with shasum $SHA1SUM"

#
# 5. Transfer backup/export + SHA1 hash file to SCP server.
#
echo `date +%H:%M` "----### Copying $BACKUP_FILE and $BACKUP_SHA1_HASH via SCP to
$SCP_SERVER ###---"
scp -o StrictHostKeyChecking=no $BACKUP_FILE $SCP_USERNAME@$SCP_SERVER:$SCP_PATH
scp -o StrictHostKeyChecking=no $BACKUP_SHA1_HASH $SCP_USERNAME@$SCP_SERVER:$SCP_PATH

# --## For FTP access ##--
# you have to define used FTP_* variables
#ftp -n $FTP_SERVER <<EOC
#quote user $FTP_USERNAME
#quote pass $FTP_PASSWORD
#binary
#debug
#cd $FTP_DIR
#put $BACKUP_FILE
#put $BACKUP_SHA1_HASH
#bye
#EOC

#
# 6. Change backup file permissions
#
echo `date +%H:%M` "----### Changing backup file permissions ###---"
chmod 640 $BACKUP_FILE $BACKUP_SHA1_HASH

#
# 7. Moving files to local archive location, deleting TEMPDIR directory
#
echo `date +%H:%M` "----### Moving $BACKUP_FILE file into $ARCHIVE_DIR directory for
backup ###---"
mv $BACKUP_FILE $BACKUP_SHA1_HASH $ARCHIVE_DIR
echo `date +%H:%M` "----### Deleting $TEMPDIR directory ###---"
rm mds_restore gtar gzip
rmdir $TEMPDIR

#
# 8. Trim backup folder to last 30 days only.
#
```

## Check Point automatic MDS backup script with upload to SSH

Publikováno na serveru Security-Portal.cz (<https://www.security-portal.cz>)

---

```
echo `date +%H:%M` "---### Deleting backups older than 30 days ###---"
find $ARCHIVE_DIR -type f -mtime +30 -exec rm {} \;

#
# All done ;o]
#

echo `date +%H:%M` "---//### ALL DONE ###\\---"

#
# -- Send MDS script log file via email
#

$SENDLOG

exit_on_error
exit
```

Věřím že se bude hodit. Navíc není problém ho trochu upravit k obrazu svému a použít ho na cokoliv jiného.

Případné komentáře uvítám.

### URL článku:

<https://www.security-portal.cz/clanky/check-point-automatic-mds-mdm-backup-script-upload-ssh>

### Odkazy:

- [1] <https://www.security-portal.cz/users/cm3l1k1>
- [2] <https://www.security-portal.cz/category/tagy/check-point>
- [3] <https://www.security-portal.cz/category/tagy/gnu/linux-bsd>
- [4] <https://www.security-portal.cz/category/tagy/networks-protocols>
- [5] <https://www.security-portal.cz/category/tagy/programming>