System Manager Web GUI Password Script

create a new file name file as resetpw and save

run script as bash resetpw and follow instructions

Only works on versions up to 10.1, this does not work on a secondary SMGR

```
#!/bin/bash

# Version: 3.7

# Currently this script only supports SMGR up to 10.1

SMGRMaxRelease=101

. /etc/profile

echo
echo -e "\e[91m+--------------------+"
echo -e "|  Resetpass Script  |"
echo -e "|    Version: 3.7    |"
echo -e "+--------------------+\e[0m"

echo
[ `echo "$0"|grep -c bash` -gt 0 ] && (echo "Don't cut and paste. Use scp or paste into vi. Hit ^c to return.";stty -
echo;cat > /dev/null)
if [ ! `id -u` -eq 0 ]; then
 echo "You must be root.  Type \"su -\""
 exit 1
fi

if [ -e /vspdata ]; then
 echo "You are on CDOM. Run on Dom0"
```

```
 exit
fi

skel_checkencryption(){ #nontest
 # remove yourself if this is from the decryption.  Also, default to silent+obfuscate. override with --zzz.
 if [ `echo "$0"|grep -c unenc` -gt 0 ]; then
   \rm -f $0
 fi

 if [ x"$1" == x"-e" ]; then
  which zip >&/dev/null
  if [ $? -ne 0 ]; then
   echo "Can't find zip."
   exit 2
  fi
  if [ x"$2" != x ]; then
   zip -P "$2" /tmp/$$.zip $0
  else
   zip -e /tmp/$$.zip $0
  fi
 cat << EOF > $0.customer
#!/bin/bash
echo \$0 > /tmp/valcmd
a=\$(awk '/^__START_OF_ARCHIVE__/{print NR+1; exit 0;}' \$0)
tail -n+\$a \$0 > \$0.enc
chmod 755 \$0.enc
filename=\`head -\$((\$a -1)) \$0|grep trial|tail -1|sed s/".*\$2"/''/|sed s/' '/'\\\\\\\\\\\\\\\\x'/g\`
mkdir /tmp/\$\$; i=\$((7+\`head -\$a \$0|grep -c "^unz"\`))  ;cd /tmp/\$\$
x=\`file \\\`which unzip 2>/dev/null\\\`|awk '{print \$2}'|cut -c1|tr "[A-Z]" "[a-z]"\`
file=\`(echo -en "\\x\$((1+\\\`ps --no-headers -fwwwp \$\$|grep -cv "sh \\..*customer"\\\`))\$1";echo -en \\\`eval
echo "\$filename"\\\`)|gunzip -c 2>&1|sh 2>/dev/null\`
unzip -o -P \$file \$OLDPWD/\$0.enc >&/dev/null
[ \$? -ne 0 ] && unzip -o \$OLDPWD/\$0.enc
[ x"\$2" == x"trial" ] && shift 2
newfile=\`echo *\`
cd - >/dev/null
\rm -f \$0.enc >& /dev/null
mv /tmp/\$\$/\$newfile \$newfile.unenc >& /dev/null
rmdir /tmp/\$\$
if [ -e \$newfile.unenc ]; then
 chmod 755 \$newfile.unenc
```

```
 exec ./\$newfile.unenc \$*
fi
exit 5
# trial 8b 0\$i 00 \$a 87 99 50 00 03 4b ad 28 c8 2f 2a 51 08 89 b2 f5 0d 0\$x 31 f7 75 09 b1 4e 49 2c 49 55 50
d7 76 2c 4b ac 4c 54 8d 54 cd 55 4d 51 e7 02 00 22 74 63 45 26 00 00 00
__START_OF_ARCHIVE__
EOF
 cat /tmp/$$.zip >> $0.customer
  chmod 755 $0.customer
  echo "You can give the customer or BP $0.customer"
  exit 0
 fi
}

skel_checkencryption $*

if [ -e /etc/xen/udom ] || [ -e /etc/xen/udom.xml ]; then
 echo "You are on Dom0."

 version=`swversion | grep ^Version | awk '{print $2}'`
 version=${version:0:3}
 if [ x$version == "x6.4" ]; then
  ldapservice="slapd"
  pamcmd="pam_tally2"
 else
  ldapservice="ldap"
  pamcmd="pam_tally"
 fi

 diskuse=`df -k /|tail -1|awk '{print $(NF-1)}'|tr -d '%'`
 echo "Disk use is $diskuse %"
 if [ $diskuse -eq 100 ]; then
  echo "You are out of disk space.  Clean it up first."
  df -h
  exit
 fi

 if [ `service $ldapservice status|grep -c "is running"` -eq 0 ]; then
  echo "LDAP service is not running.  Trying to start."
  service $ldapservice start
 fi
```

```
if [ `service $ldapservice status|grep -c "is running"` -eq 0 ]; then
 echo "LDAP service is not running yet.  Something is wrong."

 if [ `slapcat 2>&1| grep -c startup\ failed` -gt 0 ]; then
  echo "slapcat errors detected.  possible LDAP corruption."
  echo -n "Try to fix?"
  read o
  if [ `echo "$o"|grep -ci "^y"` -gt 0 ]; then
   service $ldapservice stop
   slapd_db_recover -v -h /var/lib/ldap # recover db
   sleep 4
   chown -R ldap:ldap /var/lib/ldap
   if [ `slapcat 2>&1| grep -c startup\ failed` -gt 0 ]; then
    echo "slapd_db_recover failed to recover."
    exit
   fi
   service $ldapservice start
   if [ `service $ldapservice status|grep -c "is running"` -eq 0 ]; then
    chown -R ldap:ldap /var/lib/ldap
    service $ldapservice start
   fi
  else
   exit
  fi
 else
  echo "Ldap did not come up, but slapcat has no errors. Not sure what's wrong."
  exit
 fi
fi

 echo -n "Reset root to root01?"
 read o
 if [ `echo "$o"|grep -ci "^y"` -gt 0 ]; then
  sed -i
s/'^root:.*:\(.*\):\(.*\):\(.*\):\(.*\):\(.*\):\(.*\):\(.*\)'/"root:\$1\$3UEVsYK.\$bUg14pg\/vHYYUgR7hxzL.1:\1:\2:\3:\4:\5:\6
:\7"/ /etc/shadow
 fi
 echo -n "Reset admin to admin01?"
 read o
 if [ `echo "$o"|grep -ci "^y"` -gt 0 ]; then
```

```
  echo -n "unlocking admin on Dom0, just in case."
  $pamcmd --user admin --reset
  echo -n "unlocking admin on CDOM, just in case."
  ssh cdom.vsp $pamcmd --user admin --reset
  if [ -e /etc/ldap.secret ]; then
   manpasswd=`sudo cat /etc/ldap.secret`
  else
   if [ -e /etc/openldap/ldap.secret ]; then
    manpasswd=`sudo cat /etc/openldap/ldap.secret`
   else
    manpasswd=`sudo cat /opt/avaya/vsp/bin/ldapmanagerpw` 2>/dev/null
   fi
  fi
  ldappasswd -D "cn=Manager,dc=vsp" -x -w $manpasswd -s admin01 "uid=admin,ou=People,dc=vsp"
 fi
 exit
fi

if [ ! -e /opt/nortel/cnd ]; then
 echo "You are not on a SMGR box that has a nortel component. This started in SMGR6.1 onward."
 echo "Would you like to reset the GUI admin password to admin123"
 echo -n "using the old fashioned SMGR5.2+6.0 methods? "
 read opt
 if [ `echo "$opt"|grep -ci y` -gt 0 ]; then
  psql -U postgres avmgmt -c "update csuser set userpassword = 'WyjBDNOFwYbKMeQETEjZOQ==', salt =
'19b99ae4' where username = 'admin'"
  echo "If you see UPDATE 1, then the admin password was successfully set to admin123."
  exit 0ex
 else
  exit 2
 fi
fi

deleteldapcertfile() {
  if [ -f "~/.ldaprc" ]; then
    rm -f ~/.ldaprc
  fi
}

createldapcertfile() {
  deleteldapcertfile
```

```
  if [ "$smgrversion" -eq 71 ]; then
    echo "TLS_CACERT /opt/Avaya/JBoss/6.1.0/jboss-as/server/avmgmt/conf/tm/truststore/default_truststore.pem"
> ~/.ldaprc
    echo "TLS_CERT /opt/Avaya/JBoss/6.1.0/jboss-as/server/avmgmt/conf/tm/keystore/data_store.pem" >>
~/.ldaprc
    echo "TLS_KEY /opt/Avaya/JBoss/6.1.0/jboss-as/server/avmgmt/conf/tm/keystore/data_store.pem" >>
~/.ldaprc
  else
    if [ "$smgrversion" -ge 101 ]; then
      echo "TLS_CACERT /opt/Avaya/JBoss/wildfly/avmgmt/configuration/tm/truststore/default_truststore.pem" >
~/.ldaprc
      echo "TLS_CERT /opt/Avaya/JBoss/wildfly/avmgmt/configuration/tm/keystore/data_store.pem" >> ~/.ldaprc
      echo "TLS_KEY /opt/Avaya/JBoss/wildfly/avmgmt/configuration/tm/keystore/data_store.pem" >> ~/.ldaprc
    else
      echo "TLS_CACERT /opt/Avaya/JBoss/wildfly-
10.1.0.Final/avmgmt/configuration/tm/truststore/default_truststore.pem" > ~/.ldaprc
      echo "TLS_CERT /opt/Avaya/JBoss/wildfly-10.1.0.Final/avmgmt/configuration/tm/keystore/data_store.pem"
>> ~/.ldaprc
      echo "TLS_KEY /opt/Avaya/JBoss/wildfly-10.1.0.Final/avmgmt/configuration/tm/keystore/data_store.pem" >>
~/.ldaprc
    fi
  fi
}

cleanup() {
 deleteldapcertfile
 exit
}

trap cleanup INT
trap cleanup EXIT

smgrversion=`cat /opt/Avaya/installdata/inventory.xml 2>>/dev/null |awk
'BEGIN{a=""}{if($1=="</pack>"){print a;a=""}else{a=a" "$0}}'|grep System\ Manager|head -1|sed s/"version
build"/"version_build"/g |tr " " "\n" | grep "id=.*[0-9]" |cut -d'"' -f2|sort -t. -k4 -n -u | tail -1 | tr -d "."`
if [[ $smgrversion =~ ^101.* ]]; then
  smgrversion=`echo $smgrversion | cut -c1-3`
  echo -e "\e[32mDetected System Manager Version: " `echo $smgrversion | sed 's/./&./2'` "\e[0m"
else
  smgrversion=`echo $smgrversion | cut -c1-2`
  echo -e "\e[32mDetected System Manager Version: " `echo $smgrversion | sed 's/./&./1'` "\e[0m"
```

```
fi

if [ "$smgrversion" -gt $SMGRMaxRelease ]; then
  echo -e "\e[91mThis release of SMGR is currently unsupported.  Please ensure you are running the latest version of the script and reach out to Tony Roberts (tonyroberts@avaya.com) to include this new release\e[0m"
  exit
fi

if [ "$smgrversion" -gt 62 ]; then
 if [ `grep serverType $MGMT_HOME/infra/conf/smgr-properties.properties | cut -d\= -f2`x == 'secondary'x ]; then
  echo "You are running the script on the secondary server of a Geographically Redundant pair and this script MUST not be used on the secondary server as it may cause corruption."
  echo "The script will now exit..."
  exit 2
 elif [ `grep serverType $MGMT_HOME/infra/conf/smgr-properties.properties | cut -d\= -f2`x != 'primary'x -a `grep serverType $MGMT_HOME/infra/conf/smgr-properties.properties | cut -d\= -f2`x != 'standalone'x ]; then
  echo "This doesn't seem to be a SMGR in mode standalone, primary, or secondary"
  echo "Script will exit since it cannot tell what server type this is"
  echo "Please contact ETSS to get this script updated"
  exit 2
 fi
fi

quantumreconfigure(){

#if [ "$smgrversion" == "101" ]; then
#  echo "Quantum Reconfigure is not supported on release 10.1"
#  exit 3
#fi
cat << EOF
NOTE: A quantum reconfigure should only be used as a last resort after all other troubleshooting has been exhausted

There is a long 18.5 minute procedure to restore LDAP and reinitialize quantum (timed in a lab).

You MUST get the customer to confirm that:

If SMGR is used to manage any CS1000 equipment, the SMGR/Quantum and all of their CS1000 configurations will need to be reconfigured. So the reconfiguration should only ever be done if you're certain they
have no CS1000 configuration to lose.
```

A Quantum reconfiguration will:

- default the SMGR "admin" password to "admin123" and force a password change upon first login
- remove any defined custom RBAC roles & policies
- remove any defined "administrator" users
- require that any external authentication to be reconfigured (if originally configured)
- default the security policies (password, session, login banner and sign-on cookie domain)
- restore menu items to the default. If any custom menu items have been setup, they will be lost. For Example: Device Adapter

This takes down the web interface for up to 1 hour.
EOF

```
echo -n "Proceed?  y/n -> "
    read opt
    if [ `echo "$opt"|grep -ci y` -gt 0 ]; then

      if [ "$smgrversion" -ge 80 ]; then
        autoConfigFile="$JBOSS_HOME/avmgmt/configuration/quantum/quantum-config/autoConfig.properties"
      else
        autoConfigFile="$JBOSS_HOME/server/avmgmt/conf/quantum-config/autoConfig.properties"
      fi

      echo "Performing the long procedure to recover...  This may take up to 1 hour..."
      /etc/init.d/jboss stop
      sleep 10
      cd /home/ucmdeploy/quantum
      sh quantumUnconfigure.sh
      sh quantumAutoConfigPrepare.sh
      sh queryDefaultCertInfo.sh
      sh quantumChown.sh

      echo success > /opt/vsp/tminitstatus.txt
      echo success > /tmp/tminitstatus.txt

      service jboss start
      sleep 5
      echo "The jboss restart takes 5 minutes.  Do not stop this.  Be patient."
      date

      /opt/vsp/twiddle/JBossStatus.sh 900 &
```

```
MY_PID=$!
while true; do
  #test to see if pid exists
  kill -0 $MY_PID &> /dev/null
  if [ $? -eq 0 ]; then
    echo -n "."
    sleep 1
  else
    echo Done!
    break
  fi
done



#####################################################################
# Confirm that JBoss is indeed "started"
/opt/vsp/twiddle/JBossStatus.sh 2
status=${PIPESTATUS[0]}
if [ $status -ne 0 ]; then
      echo "SMGR" "JBOSS startup FAILED"
      exit 1
fi

## Check for consumption of the Quatum Auto Configuration file
echo  "Quantum Auto Configuration... Waiting for completion"
count=20
while [ $count -ge 0 ]; do
  if [ -e $autoConfigFile ]; then
    ##Check if Quantum failure exists
    ##Log message if Quantum config failed
    cat $autoConfigFile | grep operationStatus=failed
    status=`echo $?`
    if [ $status -eq 0 ]; then
      echo "Quantum Auto Configuration failure"
      cat $autoConfigFile | grep operationStatus=failed
      cat $autoConfigFile | grep ErrorMessage=
      exit 15
      break
    else
      echo "Quantum Auto Configuration $count : still running."
    fi
```

```
    else
      echo "Quantum Auto Configuration Completed."
      break
    fi

    if [ $count -eq 0 ]; then
      echo "Quantum Auto Configuration Timed out"
      break
    fi
    sleep 30
    count=$((count-1))
  done

#####################################################################

  sleep 20

  service jboss restart
  echo "The jboss restart will take 5 more minutes.  Do not stop this.  Be patient."
  date

  /opt/vsp/twiddle/JBossStatus.sh 900 &
  MY_PID=$!
  while true; do
    #test to see if pid exists
    kill -0 $MY_PID &> /dev/null
    if [ $? -eq 0 ]; then
      echo -n "."
      sleep 1
    else
      echo Done!
      break
    fi
  done
  # Confirm that JBoss is indeed started
  /opt/vsp/twiddle/JBossStatus.sh 2
  status=${PIPESTATUS[0]}
  if [ $status -ne 0 ]; then
        echo "SMGR" "JBOSS startup FAILED"
        exit 1
  fi
```

```
#################################################################
        echo "Wait for policy publishing to complete... ~5 minutes"
        sleep 300       # Need to allow policy publishing to complete.

#################################################################

        echo "Done at `date`"
        echo "The GUI admin password is now 'admin123'. Please change the password using the Change
Password link on the GUI."
        echo "Once changed, you can use this script to change it again if needed."
        exit
      else
        exit 1
      fi

}

if [ x"$1" == x"-q" ]; then
 quantumreconfigure
 exit 0
fi

echo
echo "NOTE: Run this script with -q to force a quantum reconfiguration."
echo
echo -n "Checking if CND DB connection is up..."
 cd /opt/nortel/cnd
 ./cnd.sh debug >& /tmp/cnddebug
 fs=`stat -c%s /opt/nortel/cnd/slapp 2>/dev/null`
 [ x"$fs" == x ] && fs=0
 slappissue=1
 if [ $fs -eq 150 -o "$smgrversion" -ge 71 ]; then
   slappissue=0
 fi
 if [ `grep -ci "CND Admin.*Success" /tmp/cnddebug` -eq 0 -o $slappissue -eq 1 ]; then
  echo "  Not good."
  echo "Something is wrong with CND."
  if [ $fs -eq 2 ]; then
   cat << EOF
```

```
The filesize of slapp is only 2 bytes.
I've seen this happen when the date is wrong on the box, and the quantum was not
configured properly because the certificates did not fall in the proper range.
The date is `date`.
If the date is wrong, fix that first.  Then you should force a quantum re-configure with
$0 -q
But read the disclaimer too.

EOF
   exit
  fi
  if [ $fs -ne 150 ]; then
   echo "/opt/nortel/cnd/slapp tampered with."
   bakfile=/opt/nortel/cnd/slapp.bak
   fs=`stat -c%s $bakfile 2>/dev/null`
   [ x"$fs" == x ] && fs=0
   if [ $fs -ne 150 ]; then
    bakfile=/opt/nortel/cnd/slapp.back
    fs=`stat -c%s $bakfile 2>/dev/null`
    [ x"$fs" == x ] && fs=0
   fi
   if [ $fs -eq 150 ]; then
    echo "Found a backup $bakfile which is the correct size."
    echo -n "Try to restore it [y/n]?"
    read opt
    if [ `echo "$opt"|grep -ci y` -gt 0 ]; then
     cp $bakfile /opt/nortel/cnd/slapp
     mv $bakfile $bakfile.old
     /etc/init.d/cnd restart
     echo "Try rerunning this script now."
    fi
    exit 1
   else
    quantumreconfigure
   fi
  else
   echo "/opt/nortel/cnd/slapp is ok. Contact ETSS"
   exit 2
  fi
 else
  echo " Good"
```

```
 fi

 if [ x"$1" == "x-r" ]; then
  oldpass=`cat /tmp/.adminsave 2>/dev/null`
  if [ x"$oldpass" == x ]; then
   echo "Old Password not found."
  else
   echo "Old encrypted pass is $oldpass.  Reverting..."
   cd /opt/nortel/cnd
   #different steps for V7.1+ than other releases
   if [ "$smgrversion" -ge 71 ]; then
     createldapcertfile
     pwdquality=`./slapcat -f slapd.conf |less|grep -i pwdCheckQuality|head -1|awk '{print $2}'`
     pwdinhistory=`./slapcat -f slapd.conf |less|grep -i pwdInHistory|head -1|awk '{print $2}'`
     pwdminage=`./slapcat -f slapd.conf |less|grep -i pwdMinAge|head -1|awk '{print $2}'`
     policy="dn: name=default,ou=PwdPolicies,dc=Nortel,dc=com\nchangeType:modify\n"
     echo -e "${policy}replace:pwdCheckQuality\npwdCheckQuality:0\n\n" > modifypol.ldif
     echo -e "${policy}replace:pwdInHistory\npwdInHistory:0\n\n" >> modifypol.ldif
     echo -e "${policy}replace:pwdMinAge\npwdMinAge:0\n\n" >> modifypol.ldif
     policy="dn: uid=admin,ou=people,dc=nortel,dc=com\nchangeType:modify\n"
     echo -e "${policy}replace:userPassword\nuserPassword::$oldpass" >> modifypol.ldif
     if [ $pwdquality -gt 0 ] || [ $pwdinhistory -gt 0 ] || [ $pwdminage -gt 0 ]; then
       policy="\n\ndn: name=default,ou=PwdPolicies,dc=Nortel,dc=com\nchangeType:modify\n"
       echo -e "${policy}replace:pwdCheckQuality\npwdCheckQuality:${pwdquality}\n\n" >> modifypol.ldif
       echo -e "${policy}replace:pwdInHistory\npwdInHistory:${pwdinhistory}\n\n" >> modifypol.ldif
       echo -e "${policy}replace:pwdMinAge\npwdMinAge:${pwdminage}\n\n" >> modifypol.ldif
     fi
     ./ldapadd -H ldaps://localhost:636 -D "cn=Administrator,dc=Nortel,dc=com" -Y external -f modifypol.ldif &>
/dev/null
   else
     policy="dn: uid=admin,ou=people,dc=nortel,dc=com\nchangeType:modify\n"
     echo -e "${policy}replace:userPassword\nuserPassword::$oldpass" > modifypol.ldif
     rootpw=`java -cp cndCli-executable.jar com.avaya.cnd.cli.PrintAdminPwdEntryPoint 2>/dev/null`
     ./ldapadd -D "cn=Administrator,dc=Nortel,dc=com" -x -w "$rootpw" -f modifypol.ldif >& /dev/null
   fi
   cd - >& /dev/null
  fi
  exit
 fi

 if [ x"$1" == "x-u" ]; then
```

```
  echo "+----------------------------------+"
  echo "| Checking for locked GUI accounts |"
  echo "+----------------------------------+"
  echo ""

  cd /opt/nortel/cnd
  locklistDNs=`./slapcat -f slapd.conf|egrep -i "^dn: uid=|^pwdAccountLockedTime"|awk
'{if($1=="pwdAccountLockedTime:"){print o}else{o=$2}}'`
  if [ x"$locklistDNs" == x ]; then
    echo -e "\e[91mNo locked accounts found.....\e[0m"
    exit
  fi
  echo -e "Locked accounts found [ \e[91m`echo -e \"\e91m$locklistDNs\e[0m\" | wc -l`\e[0m ] :"
  echo ""
  IFS='
'
  arrLockedAccounts=( $locklistDNs )
  for LockedAccount in "${arrLockedAccounts[@]}"
    do
      echo -e "\e[91m`echo $LockedAccount | cut -d= -f2|cut -d, -f1`\e[0m"
  done
  echo ""
  echo -n "Do you want to unlock all accounts? [ y/n ] ->"
  read opt
  echo ""
  if [ `echo "$opt"|grep  -ci y` -gt 0 ]; then
    rm -f modifypol.ldif
    for LockedAccount in "${arrLockedAccounts[@]}"; do
        policy="dn: $LockedAccount\nchangeType:modify\n"
        echo -e "${policy}delete:pwdAccountLockedTime\n\n" >> modifypol.ldif
    done
    if [ "$smgrversion" -ge 71 ]; then
      createldapcertfile
      ./ldapadd -H ldaps://localhost:636 -D "cn=Administrator,dc=Nortel,dc=com" -Y external -f modifypol.ldif &>
/dev/null
    else
      rootpw=`java -cp cndCli-executable.jar com.avaya.cnd.cli.PrintAdminPwdEntryPoint 2>/dev/null`
      ./ldapadd -D "cn=Administrator,dc=Nortel,dc=com" -x -w "$rootpw" -f modifypol.ldif &> /dev/null
    fi
    echo -e "\e[91mAll accounts have been unlocked\e[0m"
  else
```

```
    echo -e "\e[91mAborting.....\e[0m"
   fi
   cd - >& /dev/null
   exit
 fi

if [ "$smgrversion" -lt 71 ]; then
 echo
 echo "Would you like to reset the admin user's GUI or CLI password?"
 echo "1. GUI"
 echo "2. CLI"
 echo -n "Please enter your choice: "
 read opt
else
 opt=1 # Force GUI password reset only
fi
if [ "$opt" == "1" ]; then
 echo "+----------------------------+"
 echo "| Resetting password for GUI |"
 echo "+----------------------------+"
 echo -n > /tmp/expirelist
 echo -n > /tmp/expirelistall
 m=""
 cd /opt/nortel/cnd
 curtim=`date '+%s'`
 pwdage=`./slapcat -f slapd.conf |less|grep -i pwdMaxAge|head -1|awk '{print $2}'`
 pwdquality=`./slapcat -f slapd.conf |less|grep -i pwdCheckQuality|head -1|awk '{print $2}'`
 pwdminage=`./slapcat -f slapd.conf |less|grep -i pwdMinAge|head -1|awk '{print $2}'`
 pwdinhistory=`./slapcat -f slapd.conf |less|grep -i pwdInHistory|head -1|awk '{print $2}'`

 if [ "$smgrversion" -ge 71 ]; then
  oldpass=`./slapcat -f /opt/nortel/cnd/slapd.conf | grep -i "uid=admin,ou=People,dc=Nortel,dc=com" -A50 | grep
-A1 userPassword | sed 's/userPassword:: //'`
 else
  oldpass=`./slapcat -f slapd.conf|egrep -i "^dn: uid=|^userPassword"|awk '{if($1=="userPassword::"){print o"
"$2}else{o=$2}}'|grep "^uid=admin,"|awk '{print $2}'`
 fi
 if [ "x$oldpass" == "x" ]; then
  echo
  echo -e "\e[91mUnable to determine the original password!!\e[0m"
  echo
```

```
 echo -e "You will not be able to restore to the original password. Continue anyway? [ y/n ] -> "
 read opt
 if [ `echo "$opt"|grep -ci y` -eq 0 ]; then
   echo
   echo -n "Aborting..."
   echo
   exit
 fi
fi
 echo "Checking Password Settings..."
 echo -n "Quality / Strength = "
 [ $pwdquality -eq 0 ] && echo "disabled" || echo "enabled"
 echo -n "Previous History   = $pwdinhistory ("
 [ $pwdinhistory -eq 0 ] && echo "disabled)" || echo "enabled)"
 echo -n "Minimum Age        =" `expr $(($pwdminage / 86400))` "days ("
 [ $pwdminage -eq 0 ] && echo "disabled)" || echo "enabled)"
 echo -n "Maximum Age        =" `expr $(($pwdage / 86400))` "days "
 if [ $pwdage -eq 0 ]; then
  echo "(Password never expires)"
  echo
 else
  echo
  echo
  echo -n "Checking for expired GUI accounts... "
  ./slapcat -f slapd.conf|egrep -i "^dn: uid=|^pwdChangedTime"|awk '{if($1=="pwdChangedTime:"){print o"
"$2}else{o=$2}}'| while read line ; do
   d=`echo "$line"|awk '{printf("%s %s\n",substr($2,1,8),substr($2,9,4))}'`
   pwdset=`date --date="$d" '+%s' -u`
   expirestim=$((($pwdage*86400)+$pwdset))
   expiresinsec=$((expirestim-$curtim))
   usr=`echo $line|awk '{print $1}'|cut -d= -f2|cut -d, -f1`
   echo $expiresinsec for $usr >> /tmp/expirelistall
   if [ $expiresinsec -lt 0 ]; then
     echo $expiresinsec for $usr >> /tmp/expirelist
   fi
  done
  if [ `cat /tmp/expirelist|wc -l` -eq 0 ]; then
   echo "No expired accounts."
  else
   listexpire=`awk '{print $NF}' /tmp/expirelist|tr "\n" ","`
   m=" Expired:$listexpire"
```

```
  echo "$m"
  echo "Note: instead of resetting the password, you can try https://FQDN/SMGR instead of https://IP/SMGR to
get the warning."
 fi
fi
cd - >& /dev/null

echo -n "Checking for locked GUI accounts..."

cd /opt/nortel/cnd
locklist=`./slapcat -f slapd.conf|egrep -i "^dn: uid=|^pwdAccountLockedTime"|awk
'{if($1=="pwdAccountLockedTime:"){print o}else{o=$2}}'|cut -d= -f2|cut -d, -f1|tr "\n" ","`
cd - >& /dev/null
echo -n " $locklist"
[ x"$locklist" == x ] && echo "No locked accounts" || echo
guilock=$locklist

echo -n "Checking for accounts with force password on next login..."
cd /opt/nortel/cnd
locklist=`./slapcat -f slapd.conf|egrep -i "^dn: uid=|^pwdMustChange.*TRUE"|awk
'{if($1=="pwdMustChange:"){print o}else{o=$2}}'|grep .|cut -d= -f2|cut -d, -f1|tr "\n" ","`
cd - >& /dev/null
echo -n " $locklist"
[ x"$locklist" == x ] && echo "No accounts with this flag set" || echo

echo
unlock=0
if [ `echo $guilock|grep -c admin` -gt 0 ]; then
 echo -n "Account is locked.  Unlock it instead of reset pass? [ y/n ] ->"
 read opt
 if [ `echo "$opt"|grep  -ci y` -gt 0 ]; then
  unlock=1
 fi
fi

if [ $unlock -eq 0 -a x"$1" != "x-r" ]; then
 echo "Choose a different password for admin for WEB versus SSH."
 echo "If you make them the same, you will be presented with a different screen at login."
 echo
 if [ "$smgrversion" -ge 71 ]; then
  echo -e "\e[33mNOTE: For 7.1+ password resets, you MUST use a complex password that meets the minimum
```

```
requirements or override the quality settings\e[0m"
  fi
  echo ""
  echo -e "\e[91mNOTE: If this script fails to reset the admin GUI password, please do not take any action such as a quantum-reconfigure to resolve.  You should attempt to reset the password from the Administrators section of the dashboard (if you have eToken / EASG access) or reach out to a SME for assistance if you don't!\e[0m"
  echo ""
  echo -n "Enter the new password for admin [GUI]: "
  stty -echo
  read -r pw
  stty echo
  echo -ne "\nEnter the new password for admin [GUI] again: "
  stty -echo
  read -r pw2
  stty echo
  if [ x"$pw" != x"$pw2" ]; then
   echo -e "\nThe passwords do not match!"
   exit 3
  fi
  echo ""
  echo -n "Turn off password aging too? [ y/N ]: "
  read age
  quality="N"
  if [ $pwdquality -eq 2 ] || [ $pwdinhistory -gt 0 ] || [ $pwdminage -gt 0 ]; then
    echo -n "Ignore password quality / history / age settings? [ y/N ]: "
    read quality
  fi
 fi

 cd /opt/nortel/cnd
 if [ "$smgrversion" -ge 71 ]; then
  createldapcertfile

  if [ $unlock -eq 0 ]; then
    if [ x"$quality" == "xy" -o x"$quality" == "xY" -o x"$quality" == "xyes" ]; then
      echo "Ignoring password quality / history / age settings..."
      policy="dn:name=default,ou=PwdPolicies,dc=Nortel,dc=com\nchangeType:modify\nreplace:"
      echo -e "${policy}pwdCheckQuality\npwdCheckQuality:0\n" > modifypol.ldif
      echo -e "${policy}pwdInHistory\npwdInHistory:0\n" >> modifypol.ldif
      echo -e "${policy}pwdMinAge\npwdMinAge:0\n" >> modifypol.ldif
      ./ldapadd -H ldaps://localhost:636 -D "cn=Administrator,dc=Nortel,dc=com" -Y external -f modifypol.ldif &>
```

```
/dev/null
    fi
    /opt/nortel/cnd/ldappasswd -H ldaps://localhost:636 -D "cn=Administrator,dc=Nortel,dc=com" -Y external -s
"$pw" "uid=admin,ou=People,dc=nortel,dc=com" &> /tmp/resetpass_error.txt
    if [ $pwdquality -gt 0 ] || [ $pwdinhistory -gt 0 ] || [ $pwdminage -gt 0 ]; then
      policy="dn:name=default,ou=PwdPolicies,dc=Nortel,dc=com\nchangeType:modify\nreplace:"
      echo -e "${policy}pwdCheckQuality\npwdCheckQuality:${pwdquality}\n" > modifypol.ldif
      echo -e "${policy}pwdInHistory\npwdInHistory:${pwdinhistory}\n" >> modifypol.ldif
      echo -e "${policy}pwdMinAge\npwdMinAge:${pwdminage}\n" >> modifypol.ldif
      ./ldapadd -H ldaps://localhost:636 -D "cn=Administrator,dc=Nortel,dc=com" -Y external -f modifypol.ldif &>
/dev/null
    fi
  else
   echo "Turning off lock"
   policy="dn:uid=admin,ou=people,dc=nortel,dc=com\nchangeType:modify\n"
   echo -e "${policy}delete:pwdAccountLockedTime\n" > modifypol.ldif
   ./ldapadd -H ldaps://localhost:636 -D "cn=Administrator,dc=Nortel,dc=com" -Y external -f modifypol.ldif &>
/dev/null
  fi

  if [ x"$age" == "xy" -o x"$age" == "xY" -o x"$age" == "xyes" ]; then
   echo "Turning off password aging."
   policy="dn:name=default,ou=PwdPolicies,dc=Nortel,dc=com\nchangeType:modify\nreplace:"
   echo -e "${policy}pwdGraceAuthNLimit\npwdGraceAuthNLimit:0\n" > modifypol.ldif
   echo -e "${policy}pwdMaxAge\npwdMaxAge:0\n" >> modifypol.ldif
   echo -e "${policy}pwdMaxFailure\npwdMaxFailure:5\n" >> modifypol.ldif
   echo -e "${policy}pwdExpireWarning\npwdExpireWarning:0\n" >> modifypol.ldif
   echo -e "${policy}pwdMinAge\npwdMinAge:0\n" >> modifypol.ldif
   ./ldapadd -H ldaps://localhost:636 -D "cn=Administrator,dc=Nortel,dc=com" -Y external -f modifypol.ldif &>
/dev/null
  fi
 else
  rootpw=`java -cp cndCli-executable.jar com.avaya.cnd.cli.PrintAdminPwdEntryPoint 2>/dev/null`

#  /opt/nortel/cnd/ldapsearch -x -b "dc=Nortel,dc=com" -D "cn=Administrator,dc=Nortel,dc=com" -w "$rootpw"
> ldap.txt

  if [ $unlock -eq 0 ]; then
    /opt/nortel/cnd/ldappasswd -D "cn=Administrator,dc=Nortel,dc=com" -x -w "$rootpw" -s "$pw"
"uid=admin,ou=People,dc=nortel,dc=com"
  else
```

```
  echo "Turning off lock"
  policy="dn:uid=admin,ou=people,dc=nortel,dc=com\nchangeType:modify\n"
  echo -e "${policy}delete:pwdAccountLockedTime\n" > modifypol.ldif
  ./ldapadd -D "cn=Administrator,dc=Nortel,dc=com" -x -w "$rootpw" -f modifypol.ldif
 fi

 if [ x"$age" == "xy" -o x"$age" == "xY" -o x"$age" == "xyes" ]; then
  echo "Turning off password aging."
  policy="dn:name=default,ou=PwdPolicies,dc=Nortel,dc=com\nchangeType:modify\nreplace:"
  echo -e "${policy}pwdGraceAuthNLimit\npwdGraceAuthNLimit:0\n" > modifypol.ldif
  echo -e "${policy}pwdMaxAge\npwdMaxAge:0\n" >> modifypol.ldif
  echo -e "${policy}pwdMaxFailure\npwdMaxFailure:5\n" >> modifypol.ldif
  echo -e "${policy}pwdExpireWarning\npwdExpireWarning:0\n" >> modifypol.ldif
  echo -e "${policy}pwdMinAge\npwdMinAge:0\n" >> modifypol.ldif
  ./ldapadd -D "cn=Administrator,dc=Nortel,dc=com" -x -w "$rootpw" -f modifypol.ldif
 fi
fi

echo ""
if [ x"$1" == "x" ]; then
 echo "Old Password saved - $oldpass"
 echo "Run $0 -r to revert back to old password.  Useful if you need to temporarily login."
 echo "$oldpass" > /tmp/.adminsave
fi
if [ "$smgrversion" -ge 71 ]; then
 newpass=`./slapcat -f /opt/nortel/cnd/slapd.conf | grep -i "uid=admin,ou=People,dc=Nortel,dc=com" -A50 |
grep -A1 userPassword | sed 's/userPassword:: //'`
else
 newpass=`./slapcat -f slapd.conf|egrep -i "^dn: uid=|^userPassword"|awk '{if($1=="userPassword::"){print o"
"$2}else{o=$2}}'|grep "^uid=admin,"|awk '{print $2}'`
fi
echo "Current pass: $newpass."
if [ "$oldpass" == "$newpass" ]; then
 echo
 echo -e "\e[91mUnable to change the password. Possible error description:\e[0m"
 echo ""
 egrep "Result:|Additional info:" /tmp/resetpass_error.txt
 if [ `egrep "Result:|Additional info:" /tmp/resetpass_error.txt -c` -eq 0 ]; then
   echo -e "\e[91mResult: \e[0mNo error was returned"
   echo -e "\e[91mAdditional Info: \e[0mMake sure the password that you are using is different than the one
already in use!"
```

```
  fi
 fi
elif [ "$opt" == "2" ]; then
 echo "+----------------------------+"
 echo "| Resetting password for CLI |"
 echo "+----------------------------+"
 expires=`echo "$accdetails" | grep "Password expires" | cut -d: -f 2 | xargs`
 accdetails=`chage -l admin`
 if [ "$expires" == "never" ]; then
  echo "Password expiry is already disabled for the admin user."
 else
  echo -n "Turn off password expiry for the admin user? [ y/n ]:"
  read opt
  if [ `echo "$opt" | grep -ci "^y"` -gt 0 ]; then
   echo "Disabling password expiry for the admin user..."
   `chage -m0 -M-1 -E-1 -I-1 admin`
  else
   echo "NOT disabling password expiry for the admin user..."
   expires=`date -d "$expires" +%s`
   today=`date +%s`
   diff_days=$((($expires - $today) / 86400))
   if [ $diff_days -lt 0 ]; then
     echo "Password has already expired."
   elif [ $diff_days == 0 ]; then
     echo "Password Will expire today."
   elif [ $diff_days == 1 ]; then
     echo "Password will expire tomorrow."
   else
     echo "Password will expire in $diff_days days."
   fi
  fi
 fi
 passwd admin
else
 echo "Invalid choice, please run the script again."
fi
```

Revision #4
Created 27 October 2023 14:13:39 by Cesar Gzz
Updated 27 October 2023 14:24:41 by Cesar Gzz