VM Launch Script

The following is a script (with GUI interaction via "zenity") to automate the execution of the VDD VMs. The script let a user choose the desired distro (operating system), the desktop environment (obviously not the case for Windows) and, if it's the first time he launch it, let a user also choose to get or not an encrypted share folder. Furthermore, if it's the case, it let the user to enter its own passphrase to crypt and/or decrypt that share folder.

So the script works as an automated way of running many of the tasks outlined in previous sections (in particular many steps of Xephyr and Privacy sections). Nevertheless be aware that to correctly execute the script you still need to do some preliminary steps, as like as installation of needed packages and some configurations, some of them not yet outlined before.

Here a sum of the most important steps you need to do, before to use the script. Of course we also assume here that you already have a working vdd-server, id est an ssh-able server with working xen, ltsp and running virtual machines, each equipped with at least one working complete desktop environment and ssh-server too. If needed, see previous section for details.

Requirements on the vdd server to correctly run the GUI launch script

Packages

- xephyr and rdesktop must be installed

- dmsetup, cryptsetup, samba and libpam-mount must be installed; aes and crypt modules must be loaded

- zenity must be installed on vdd-server

Users

- you need to have at least one normal user both on vdd/ltsp server and on each virtual machine you want to run

- the users on the vdd server must be "sudoers" and authorized to "sudo" on some commands without password;

this is our vdd server /etc/sudoers file last line

```
%sudo ALL=NOPASSWD: /bin/mount, /bin/chown, /sbin/lvcreate, /sbin/cryptsetup, /sbin/mke2fs, /sbin/mount.crypt, /sbin/umount.crypt, /bin/mkdir
```

- the users on the vdd server must be samba users too ("smbpasswd -a user" must be executed for each user)

- the users on the vdd server should be at least members of the following groups: adm sudo audio video plugdev netdev powerdev fuse sambashare

SSH conf

- you need ssh-keys exchange between each normal user on vdd/ltsp server and the corresponding normal user on each virtual machine you want to run; this will make the ssh connection work without a password being requested; look on line for some how-to to arrange that

- you need to edit /etc/ssh/ssh_conf on vdd-server to let "XCH" and "DE" vars being sent to virtual machines

```
SendEnv XCH DE
```

- you need to edit /etc/ssh/sshd_conf on each virtual machine to let the ssh server accept the "XCH" and "DE" vars

AcceptEnv XCH DE **Notes to modify the script**

The script steps are commented out to easy modify them to satisfy anyone's needs. In particular probably you'll have to modify at least:

- zenity argument to match YOUR virtual machines

- "VG" (volume group) var to match YOUR volume group name

#!/bin/sh #/usr/bin/launchgui # ********** # Copyright notice # # (c) 2009 Binario Etico Soc. Coop. info(@)binarioetico.org # All rights reserved # # This script is part of the VDD-Project www.vdd-project.org. This script is # free software; you can redistribute it and/or modify # it under the terms of the GNU General Public License as published by # the Free Software Foundation; either version 2 of the License, or # (at your option) any later version. # # The GNU General Public License can be found at # http://www.gnu.org/copyleft/gpl.html # # This script is distributed in the hope that it will be useful, # but WITHOUT ANY WARRANTY; without even the implied warranty of # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the # GNU General Public License for more details. # # This copyright notice MUST APPEAR in all copies of the script! # Author: Fabrizio Nasti fabrizio.nasti(@)binarioetico.org # ********** # ### This script is intended to be executed by a user on a LTSP thin client connected to the ### VDD/LTSP server (XEN Dom0) to launch para- or fully virtualized user Desktop Environments ### on already running Xen DomUs. In details it is intended to: ### 1. launch an xnested environment via Xephyr ### 2. access via ssh the desired virtual machine (Xen DomU) and start the desired Desktop ### Environment ### or to: ### 1-2. start a remote desktop connection on a Windows XP / Windows 7 virtual machine ### This script is also intended to: ### 3. create and/or activate an lvm-based encrypted or non-encrypted per-user share folder. ### ### The script use "zenity" to provide a graphical user dialog interface to ### enter the desired following parameters: distro, desktop environment, crypted or not crypted share ### folder, passphrase to crypt and de-crypt the share folder. # # sleep 2 # ## Choose the distro (operating system) # VM=$(zenity --width=250 --height=300 --list --title="Operating System" --text="Choose your Virtual Operating System"
--radiolist --column "Choose" --column "OS" --column "VM" FALSE "Windows XP" winxpvm FALSE "Windows XP 2"
winxpvm2 FALSE "Windows 7" win7vm FALSE "Debian Lenny" lennyvm FALSE "Ubuntu Jaunty" jauntyvm FALSE
"Centos5" centos5vm FALSE "Fedora11" fedora11vm --hide-column=3 --print-column=3) echo $VM case $VM in "") exit 1 ;; [win]*) echo -e "No Desktop Environment to choose"; ;; *) # ## Choose the desktop environment # DE=$(zenity --width=250 --height=300 --list --title="Desktop Environment" --text="Choose your Desktop Environment"

```
 --radiolist --column "Choose" --column "Desktop" --column "DE" TRUE KDE3/4 startkde FALSE
Gnome gnome-session FALSE
 XFCE xfce4-session --hide-column=3 --print-column=3)   if [ -z "$DE" ]; then    exit 1   fi   ;;
esac  #  ## Choose, create and activate encrypted or non-encrypted per-$USER share   #
VG=serv1  #  if [ -e /dev/$VG/"$USER"_enc ]; then   echo "encrypted device is already existing"
 if [ -e /home/$USER/share ]; then    echo "the folder share is already existing"    else    echo
"...creating the folder share"    mkdir /home/$USER/share   fi   if [ -e
/var/lib/samba/usershares/"$USER"_share ]; then    echo "the share is already active"    else
echo "...activating the share"    net usershare add "$USER"_share /home/$USER/share
"$USER share" "$USER":F   fi   if (mount | grep "$USER"_enc 1> /dev/null); then    echo "the
encrypted device is already mounted on /home/$USER/share"    else    MNT=0    while [ $MNT
-eq 0 ]; do    if zenity --entry --title="DE-ENCRYPTION PASSPHRASE" --text="Insert the
passphrase to decrypt your share folder"
        --hide-text 1> /home/$USER/passphrase; then      if (sudo mount.crypt
/dev/$VG/"$USER"_enc /home/$USER/share < /home/$USER/passphrase) 1> /dev/null; then
 MNT=1     fi     rm -f /home/$USER/passphrase     sudo chown -R $USER.$USER
/home/$USER/share     else exit 1;    fi    done   fi  fi  #  if [ -e /dev/$VG/$USER ]; then   echo
"non-encrypted device is already existing"   if [ -e /home/$USER/share ]; then          echo
"the folder share is already existing"          else          echo "...creating the folder
share"          mkdir /home/$USER/share        fi   if [ -e
/var/lib/samba/usershares/"$USER"_share ]; then        echo "the share is already active"
 else        echo "...activating the share"    net usershare add "$USER"_share
/home/$USER/share "$USER share" "$USER":F        fi   if (mount | grep
/dev/mapper/"$VG"-"$USER" 1> /dev/null); then    echo "the device is already mounted on
/home/$USER/share"    else    sudo mount /dev/$VG/$USER /home/$USER/share    sudo
chown -R $USER.$USER /home/$USER/share   fi  fi  #  if [ ! -e /dev/$VG/"$USER"* ]; then
echo "device doesn't exist"   if KD=$(zenity --list --title="Data Encryption" --text="Do you want an
encrypted share folder?" --radiolist --column "Choose"
      --column Encryption --column KD TRUE NO 0 FALSE YES 1 --hide-column=3
--print-column=3); then   echo $KD   if [ $KD == 1 ]; then    echo "...creating encrypted device"
 sudo lvcreate -n "$USER"_enc -L 1G --addtag "$USER"_enc $VG    if zenity --entry
--title="ENCRYPTION PASSPHRASE" --text="Insert the passphrase to encrypt your share
folder"
        --hide-text 1> /home/$USER/passphrase; then     sudo cryptsetup --verbose -c
aes-cbc-essiv:sha256 -q luksFormat /dev/$VG/"$USER"_enc < /home/$USER/passphrase
sudo cryptsetup luksOpen /dev/$VG/"$USER"_enc "$USER"_enc < /home/$USER/passphrase
>1 /dev/null     echo "...creating filesystem"     sudo mke2fs -j /dev/mapper/"$USER"_enc     [ -e
/dev/$VG/"$USER"_enc ] && [ ! -e /home/$USER/share ] &&       mkdir /home/$USER/share     if
[ -e /var/lib/samba/usershare/"$USER"_share ]; then          echo "the share is already active"
      else          echo "...activating the share"      net usershare add "$USER"_share
/home/$USER/share "$USER share" "$USER":F          fi     sudo cryptsetup luksClose
"$USER"_enc     else exit 1;    fi    MNT=0    while [ $MNT -eq 0 ]; do    if zenity --entry
--title="DE-ENCRYPTION PASSPHRASE" --text="Insert the passphrase to decrypt your share
folder"
        --hide-text 1> /home/$USER/passphrase; then      if (sudo mount.crypt
/dev/$VG/"$USER"_enc /home/$USER/share < /home/$USER/passphrase) 1> /dev/null; then
```

```
 MNT=1      fi      rm -f /home/$USER/passphrase      sudo chown -R $USER.$USER
/home/$USER/share      else exit 1;     fi     done    else      echo "...creating non-encrypted device"
 sudo lvcreate -n $USER -L 1G --addtag $USER $VG     echo "...creating filesystem"
sudo mke2fs -j /dev/$VG/$USER     [ -e /dev/$VG/$USER ] && [ ! -e /home/$USER/share ] &&
mkdir /home/$USER/share     if [ -e /var/lib/samba/usershare/"$USER"_share ]; then             echo
"the share is already active"             else             echo "...activating the share"      net usershare
add "$USER"_share /home/$USER/share "$USER share" "$USER":F             fi     sudo mount
/dev/$VG/$USER /home/$USER/share     sudo chown -R $USER.$USER /home/$USER/share
fi    else exit 1;    fi  fi  #  ## Launch the desired Virtual Desktop  #  case $VM in  [win]*)    if [ -d
/var/www/$USER ]; then     echo VM=$VM >> /var/www/$USER/vm_$USER$VM          else
         sudo mkdir -m 757 /var/www/$USER &&     echo VM=$VM >>
/var/www/$USER/vm_$USER$VM          fi    rdesktop -f -x l $VM &> /dev/null     ;;  *)    echo $VM
# Set the X channel to be used first by 'Xephyr' and then by 'export DISPLAY'    XCH=$[`cat
/root/Xephyr_offset` +1]     # Exclude the use of XCH 10 and 11 (i don't know why but they don't
work)    if [ `echo $XCH` -eq 10 ]; then      XCH=$[$XCH +2]    fi      if [ `echo $XCH` -eq 11 ]; then
  XCH=$[$XCH +1]    fi      # Check if a Xephyr process is using the set X channel (XCH)    while
(ps axf | grep "Xephyr" | grep :$XCH 1> /dev/null); do      XCH=$[$XCH +1]      if [ `echo $XCH`
-eq 10 ]; then       XCH=$[$XCH +2]      fi      if [ `echo $XCH` -eq 11 ]; then       XCH=$[$XCH +1]
   fi   done      # Launch Xephyr on $XCH in fullscreen mode and put it in background    Xephyr
-ac :$XCH -fullscreen &      # MOD: Get the Xephyr PID e write it in an apache chroot file
XEPHYR_PID=$!    echo XEPHYR_PID=$XEPHYR_PID    if [ -d /var/www/$USER ]; then     echo
XEPHYR_PID=$XEPHYR_PID > /var/www/$USER/vm_$XCH    else      sudo mkdir -m 757
/var/www/$USER &&      echo XEPHYR_PID=$XEPHYR_PID > /var/www/$USER/vm_$XCH    fi
   # Update the first free X channel to be used (offset)    echo $XCH > /root/Xephyr_offset      #
When XCH get 100 re-set the offset to 0     if [ "$XCH" -ge "100" ]; then      echo 0 >
/root/Xephyr_offset    fi      # Export custom env variables. They will be passed through ssh to the
desired VM (according with     # server (Dom0)  /etc/ssh/ssh_config and virtual machines
(DomUs) /etc/ssh/sshd_config)    export XCH=$XCH    export DE=$DE      # the USER variable is
read from standard local environment      # Export desired VM DISPLAY towards the LTSP
thin-client (id-est the server) and start the desired desktop     # environment (DE)       ssh
$USER@$VM 'export DISPLAY=192.168.108.21:$XCH && $DE' &    # Write the script variables
in a per-$XCH file in the per-$USER Apache root director    echo XCH=$XCH >
/var/www/$USER/vm_$XCH     # Write the script variables in a per-$XCH file in the per-$USER
Apache root directory    echo VM=$VM >> /var/www/$USER/vm_$XCH    echo DE=$DE >>
/var/www/$USER/vm_$XCH    ;;  esac  #  ## If everything went fine exit without errors    #  exit 0
```

{backbutton}