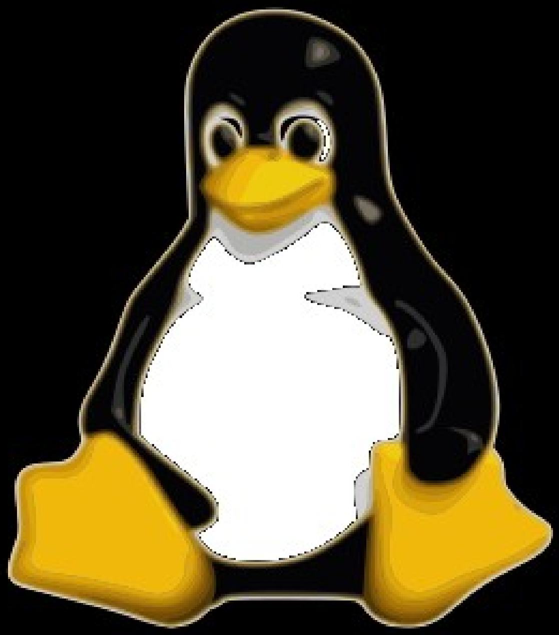
A Linux SysAdmin's Guide to Mischief



B D TASKER

The Linux SysAdmin's Guide to Mischief

B D Tasker

I object to DRM on principle, so you should never receive a copy of this book encumbered with DRM. Instead, I simply ask that you respect my rights as an author (no matter how poor) and refrain from infringing on the Copyright of my works.

If you would like to use my work in some way, please contact me at the above address to seek permission first! If for some reason the version you've received does include DRM, please let me know.

Contents

Introduction

There are numerous tricks you can play on your users, and this book attempts to detail a few of those. It's assumed that you are able to gain root level access to the user's machines, if you aren't then you are probably in an environment where playing pranks may not be a good idea! Throughout this book it's also assumed that the victim machines are running Linux with KDE (though many will work if you're running a different desktop environment).

Many of the most obvious 'pranks' you could play often involve deleting, relocating or spoofing the users files. We'll be trying to avoid that, and going for slightly more advanced techniques.

The aim is mischief, not destruction so choose your victims well. Make sure they have a sense of humour, and don't fall into the trap of singling one person out!

This book is laid out by application, so each chapter will contain minor tweaks you can make to common applications in order to make life fun! In each chapter, we'll start with basic tricks before moving to the slightly more technical.

Many of the steps will seem a little like they are over-complicating matters. Remember that the idea is that you should be able to set-up everything in this book remotely, even if the user is using the machine at the time!

All tricks in this book are aimed at workstations, as it's generally not considered a wise move to tinker with servers in this way!

Social Note: The author has been informed by someone who actually possesses social skills that it's not usually considered polite to play tricks on someone you don't know. He though it seemed wise to pass this information on, even if he doesn't entirely understand why.

Note: The author can accept no responsibility for anything that occurs as the result of you playing tricks on the wrong user, or destroying a system, or indeed anything else that may occur. He is, however, happy to accept credit for any laughs it may give you!

Disclaimer: Most of the 'technical' tricks aren't what we would normally consider technically challenging. It's just a convenient word as it's easier than saying "a trick that your users could never pull off without guidance and help"

Further Disclaimer: For those who would make the mistake of assuming that Linux boxes are insecure because of the content of this book, be aware that no machine is ever secure once you have physical access. Donations of any machine are welcome if you want to prove the Author wrong.

Final Disclaimers: The author takes no responsibility if the content of any of these disclaimers proves to be inaccurate, or if the content of the book proves to be inaccurate, or if the content of either proves to be accurate. He also takes absolutely no responsibility if anyone is stupid enough to let any part of the content fall into ~~enemy~~ users hands.

Absolutely Final Disclaimer: The author loves users really!

Pre-Requisites

You should have SSH access to the victim machine, and should be able to run commands with root privileges. Whether that means a root login, *su* or *sudo* is entirely up to the reader. It's always possible to gain root level access if you have physical access to the machine, but this book won't be telling you how to do so (wouldn't want to give the ~~victims~~ users ideas, would we?)

Common Steps

It shouldn't need to be said, but at no point should you actually *delete* anything. Unless you really want to be accused of damaging systems, it's far better to simply move files so that you have a backup to return to once you think the user has had enough!

We'll be using the variable $VICTIM a lot. This won't actually be set on your system, so substitute the username of the victim.

It's also assumed that where replacement media (sounds/video/images etc) are required, you have them stored in /root. If not, just substitute the correct path

It's also wise to check the audio volume before running anything (obviously we want it to be as loud as possible!). A word of warning, however, it'd be foolhardy not to check whether the user is already running something that may make them notice you turn the music up!

```
ps aux
# Look for anything that makes sound, Amarok, Spotify etc
# Didn't find anything? Good
alsamixer
# Use the arrow keys to navigate between controls and to adjust. Esc to quit
```

Tip: If you do find that they're running something that may be making noise, do your prep and then kill the program with pkill. In the time it takes them to go “huh?” you can have set your prank running.

KDE

An obvious and easy target is the Desktop environment itself, KDE is easily customised to achieve absolute irritation from the unfortunate ~~victim~~ user.

Desktop

An easy, if slightly puerile prank. Change the users Desktop background to something else (Perhaps a pink Justin Bieber wallpaper?).

To do so

```
cd ~$VICTIM/.kde/share/config
cat plasma-desktop-appletsrc | grep “wallpaper=”

# We should now have a line like
# wallpaper=/usr/share/wallpapers/blah.jpg
#
# This is the path to the wall paper

cd /usr/share/wallpapers/
cp blah.jpg blah.jpg.old
cp /root/Justin-Bieber-fanclubPink.jpg ./blah.jpg
chown root:root blah.jpg
```

We've now set the KDE wallpaper to use an image of the Bieber. For many users, you may find that the path to wallpapers is actually */home/$VICTIM/.kde/share/wallpapers* in which case you will want to *chown* the file to their username instead of root.

System Notifications

The KDE notifications provide a great opportunity for mischief, especially as most users don't actually know how to change them! The trick, of course, is to choose something subtle so that it's not immediately obvious.

The steps below assume that your victim hasn't changed the sounds from the default settings

```
cd /usr/share/sounds/
cp KDE-Sys-Log-In-Short.ogg  KDE-Sys-Log-In-Short.ogg.old
mv /root/Bieber-clip.ogg ./KDE-Sys-Log-In-Short.ogg
cp KDE-Sys-Log-Out.ogg KDE-Sys-Log-Out.ogg.old
mv /root/Bieber-clip2.ogg ./KDE-Sys-Log-Out.ogg
```

Now, whenever the user logs in or out our new media will play. In this case, it's a cut of a Justin Bieber song.

Warning: Your boss may or may not appreciate having Bieber sing “baby” at them! Make changes at your own risk!

You can also change a few other sounds, but remember that the key is to be subtle.

```
cd /usr/share/sounds
cp KDE-Sys-Warning.ogg KDE-Sys-Warning.ogg.old
mv /root/Homer-Simpson-Doh.ogg ./KDE-Sys-Warning.ogg
```

It's important that you ensure all audio clips are in ogg format. The system will accept other formats, but as it's expecting an ogg it's better to avoid risking an error!

AutoStarts

Just as the 'Startup' folder was a favourite for pranksters on Microsoft™ Windows®, the Autostart folder is an easy target for those using KDE.

```
cd /home/$VICTIM/.kde/Autostart
nano ImaBelieber

# Enter the following

#!/bin/bash
firefox “https://twitter.com/justinbieber”

# Ctrl-X followed by enter to exit and save

chmod +x ImaBelieber
chown $VICTIM ImaBelieber
```

You can run any command you want, just include it in the autostart file (which is, after all, just a BASH script).

A particular favourite is

```
chromium-browser --start-maximised --kiosk –login-screen-size="800,600"
“http://www.badgerbadgerbadger.com”
```

Skype

Skype gives plenty of opportunity for mischief, if only because of the number of sounds it uses!

Notifications

```
cd /usr/share/skype/sounds
cp CallRingingIn.wav CallRingingIn.wav.old
mv /root/PeanutButterJellyTime.wav CallRingingIn.wav
```

Note: You will need to ensure the WAV you used is no greater than 2MB otherwise it just won't play. Less than a 1MB is ideal.

The above example changes the sound made when someone calls the victim on Skype. You can also change the new message sound as follows

```
cd /usr/share/skype/sounds
cp ChatIncomingInitial.wav ChatIncomingInitial.wav.old
mv /root/badgerbadgerbadger.wav ChatIncomingInitial.wav
```

If you want to be **really** annoying change the files *ContactOnline.wav* and *ContactOffline.wav* to something else.

Idle Times

You can also alter some of Skype's settings, though it won't always work when Skype is already open. It'll take forever for your victim to actually notice, though, so it's not the most effective of pranks!

```
nano /home/$VICTIM/.Skype/config.xml

# Find the line <IdleTimeForAway> and change the value between the two tags (I
suggest 5!).
# The next line should be <IdleTimeForNa>. Change this as well!

cp /home/$VICTIM/.Skype/config.xml /home/$VICTIM/.Skype/config.xml.new
pkill skype && mv /home/$VICTIM/.Skype/config.xml.new /home/
$VICTIM/.Skype/config.xml

# The new settings will be loaded when they re-open Skype
```

Roaming Profiles

If your users have roaming profiles, you can take advantage of the perceived ease with which mistakes are made (given that your users may consider you incompetent – they only see you when things go wrong!)

A good way to do so, is to exploit user curiosity (with the added benefit that you can try and claim it was an informal test that documented procedures were being observed!)

```
nano /root/CONFIDENTIAL-PERSONNEL.desktop

[Desktop Entry]
Type=Application
Exec=/home/$USER/.nosyparker.sh
Icon=folder
Terminal=false
Name=CONFIDENTIAL – PERSONNEL
Comment=NOT FOR DISTRIBUTION OUTSIDE PAYROLL

# Ctrl-X followed by Enter to exit and save

chown $VICTIM  /root/CONFIDENTIAL-PERSONNEL.desktop

chmod +x /root/CONFIDENTIAL-PERSONNEL.desktop

# Don't move it yet!
```

Next we'll create the command that will be run if (when) the user tries to open our faux directory.

```
nano /home/$VICTIM/.nosyparker.sh

#!/bin/bash

# Cruel: Turn the volume right up
amixer set Master playback 100% unmute
amixer set PCM playback 100% unmute

# Run the payload
mplayer /home/$USER/.badsong.mp3 && firefox “http://www.spicegirls.co.uk”


# Ctrl-X followed by Enter to exit and save

chmod +x /home/$VICTIM/.nosyparker.sh
chown $VICTIM  /home/$VICTIM/.nosyparker.sh
```

Finally, we provide the audio and make sure all the files are where they need to be

```
cp /root/wannabe.mp3 /home/$VICTIM/.badsong.mp3
chmod 766 /home/$VICTIM/.badsong.mp3

# Finally, we need to place the Desktop file in a sensible location.
# Where is best depends on how your systems are configured.
# Users may not see files placed in ~/Desktop, their home dir may be
# a better bet

mv /root/CONFIDENTIAL-PERSONNEL.desktop /home/$VICTIM/Desktop/
```

To the user, it should appear as though the file is a directory. Given that it's labelled confidential you can guarantee a few of them will want to try and have a nose through. Attempting to do so, however, will instead load a webpage and play some music (hopefully loudly!).

Spotify

Spotify for Linux supports DBus, so we can interact with it from the CLI to make it do what we want. To do so we need to create a control script (courtesy of Ubuntu Form user Azzid)

```
nano /tmp/Spotify

#!/bin/bash

# Collect DBUS_SESSION_BUS_ADDRESS from running process
function set_dbus_adress
{
        USER=$1
        PROCESS=$2
        PID=`pgrep -o -u $USER $PROCESS`
        ENVIRON=/proc/$PID/environ

        if [ -e $ENVIRON ]
         then
        export `grep -z DBUS_SESSION_BUS_ADDRESS $ENVIRON`
         else
        echo "Unable to set DBUS_SESSION_BUS_ADDRESS."
        exit 1
        fi
}

function spotify_cmd
{
        dbus-send --print-reply --dest=org.mpris.MediaPlayer2.spotify
/org/mpris/MediaPlayer2 org.mpris.MediaPlayer2.Player.$1 1> /dev/null
}

function spotify_query
{
        qdbus org.mpris.MediaPlayer2.spotify /org/mpris/MediaPlayer2
org.freedesktop.DBus.Properties.Get org.mpris.MediaPlayer2.Player PlaybackStatus
}

function quit_message
{
        echo "Usage: `basename $0` {play|pause|playpause|next|previous|stop|playstatus|
<spotify URI>}"
        exit 1
}

# Count arguments, must be 1
if [ "$#" -ne "1" ]
then
        echo -e "You must supply exactly one argument!\n"
        quit_message
fi

# Check if DBUS_SESSION is set
if [ -z $DBUS_SESSION_BUS_ADDRESS ]
         then
        #echo "DBUS_SESSION_BUS_ADDRESS not set. Guessing."
        set_dbus_adress $VICTIM spotify
fi

case "$1" in
        play)
                spotify_cmd Play
                ;;
        pause)
                spotify_cmd Pause
                ;;
        playpause)
                spotify_cmd PlayPause
                ;;
        next)
                spotify_cmd Next
                ;;
        previous)
                spotify_cmd Previous
                ;;
        stop)
                spotify_cmd Stop
                ;;
        spotify:user:*)
                spotify_cmd "OpenUri string:$1"
                spotify_cmd Play
                ;;
        spotify:*:*)
                spotify_cmd "OpenUri string:$1"
                ;;
        playstatus)
                spotify_query
                ;;
        *)
                echo -e "Bad argument.\n"
                quit_message
                ;;
esac

exit 0


# Press Ctrl-X followed by enter to save
chmod +x /tmp/Spotify
/tmp/Spotify pause
```

This may not work on all systems, as later versions of Spotify don't communicate with dbus as well. Make sure you've substituted $VICTIM for the username of your victim too!

Festival

Most systems won't have Festival installed by default, but it's quick and easy to install and well worth it if you have users that are easily confused!

```
#*buntu
apt-get install festival

# RPM Distros

yum install festival


# Gentoo based

emerge festival
```

To use festival we simply pipe some text in

```
echo “Please don't bang my keys, I am hung over” | festival --tts
```

The victim machine will then read the text we've provided out loud. If you receive an error saying “Linux: can't open /dev/dsp” it means that the victim is running something that's using the soundcard exclusively (Spotify maybe?). There are tweaks you can do to overcome this, but it's outside the scope of this book (far too serious).

Shutdown Sounds

Now that Festival is installed, there's no reason you can't go one step further. How about adding a task to the system's shutdown parameters.

```
nano /etc/rc0.d/S01Noisy

#!/bin/bash

echo “Erasing Hard Drive” | festival --tts

# Ctrl-X followed by Enter to exit and save
chmod a+x /etc/rc0.d/S01Noisy
```

It should be the first script run as part of the shutdown procedure, which depending on the speed of your hardware means that they should still be stood by the PC!

Internet Access

One thing that really drives users wild is when their internet access is tampered with. It's perhaps one of the best areas, then, for a few pranks. Especially if you know that users shouldn't be accessing a particular site whilst at work (We'll use facebook as an example).

Blocking Sites

So, we want to create a rule to redirect those requests to somewhere else. We need to find a suitable server that won't worry that the HTTP Host header will be for Facebook.com. For the sake of example, though, we'll redirect to Google.

```
nano /etc/hosts
# Add the following line

173.194.41.160 facebook.com www.facebook.com
```

Broken Internet

For additional bonus points, install a web-server on the victims machine (Apache/LightHTTPD or whatever, your choice) and redirect all requests to a customised error page such as the one below. To do so, follow the steps above but enter 127.0.0.1 as the IP. Redirection can be achieved using a htaccess file.

```
<html>
<head>
<title>ERROR: Internet Unavailable – Automated Message</title>
</head>
<body>
<span style=”color: red”><h1>ERROR: Internet Unavailable – Automated
Message</h1></span>
<br /><br />
The Internet appears to be unexpectedly unavailable. This may be as a result of
someone entering 'Google' into the Google search engine. Engineers have been
scrambled to resolve the issue, please help us trace the fault by clicking 'Refresh'
regularly.
<br /><br />
Thank you for your patience
<br />
<i>This was an automated message from
The Internet Tractability Service</i>

</body>
</html>
```

The more effort you put into making the styling look genuine, the more likely it is you'll actually fool at least one user!

Additional Ideas

If you run a proxy server on your network, consider adding a few client specific rules to that. For example you could redirect all requests destined for “Guardian.co.uk” to “Dailymail.co.uk”. Or you could divert all requests destined for “Dailymail.co.uk” to “mailwatch.co.uk”.

It's not wise, though, to divert pages relating to strong beliefs. For example, setting a redirect from “vatican.va” to “newscientist.com” is likely to end very badly for you (and the comments section on NS will go even further downhill!).

Hardware

Sometimes all you want to do is make your users jump (how else can you be sure they're awake and working?), you may not even want them to realise that someone's actually responsible for some weird behaviour.

Thankfully, there are options available, depending entirely on what hardware is available

Basic Tricks

Sometimes the best pranks are also the easiest! Try running the following command

```
eject /dev/cdrom && sleep 2 && eject -t /dev/cdrom
```

Display

To tinker with the user's display we first need to allow ourselves access to their display. It's not too difficult to achieve if we use the users Xauthority file to gain the cookie we need

```
xauth merge /home/$VICTIM/.Xauthority
```

Next we need to find out which display X is running on. It's usually the first argument given when Xorg is loaded, so

```
ps aux | grep X
# Gives
# /usr/bin/X :0 vt7 -nr -nolisten tcp -auth /var/run/xauth/A:0-IGLBbc
export DISPLAY=:0
```

We can now run graphical applications on their desktop! (*A word of warning: at time of writing pushing firefox to the users display will mess up permissions in their home directory!*)

Rotation

Let's start by rotating the users display 180 degrees (not every system will actually be able to do this!)

```
xrandr -o inverted
```

Embarassing Pages

Although we can't risk using Firefox due to the apparent permissions issue, that doesn't prevent us from using another browser to open some embarrassing pages. (*Note: It's really not recommended that you open **anything** of an adult nature, you never know who you'll upset!*)

```
/usr/bin/konqueror “http://www.bieberfever.com”
```

Timewasting

This trick involves setting an application to open a set number of times. The application you choose depends on what you want to achieve (and what's installed and available).

For example

```
export X=0
while [ $X -lt 10 ]; do /usr/games/ksudoku; X=$(( $X + 1 )); done
```

Will only open one Sudoku window at a time, but will launch it ten times (with the next opening whenever they hit close).

You could instead opt to send them a message

```
export X=0
export MSG=”I Must not drink Ben's coffee!”
while [ $X -lt 10 ]; do echo & kdialog --passivepopup "$MSG" 10; sleep 1; X=$(( $X + 1 ));
done
```

You could of course, choose instead to pepper their desktop with something instead. To do so, you need to find an application that forks into the background or force it to do so.

```
export X=0
while [ $X -lt 10 ]; do kcalc & echo ; X=$(( $X + 1 )); done
```

Locked Screen

To lock their screen, run

```
/usr/lib/kde4/libexec/kscreenlocker --forcelock
```

BedTime

We can lock their screen, but all they need do is enter their password to unlock. If it appears that their screen has died it often takes the victim a little longer to figure out quite what's going on.

```
xset dpms force off
sleep 30;
xset dpms force off
```

As soon as they move the mouse or touch a key on the keyboard the system will wake the screen again, so we wait 30 seconds before we power the screen down again!

If you're feeling particularly evil, you could create a simple script to monitor the status of the monitor (check with xset -q) designed to switch it off again as soon as it's re-enabled. A truly evil sysadmin may forget to include a loop counter and so need to manually interrupt the script!

More Advanced Tricks

We've all seen users swap keyboards and mice between machines so that the users are unknowingly interacting with the machine next to them. However, we're Sysadmins and are completely above that!

Instead, try configuring X forwarding on both machines so that the the GUI for machine 1 displays on machine 2's monitor and vice-versa. Bonus points if you can situate yourself somewhere that you can see the users trying to work out how the input devices have been switched!

Combining Techniques

The more advanced prankster will recognise that sometimes it can be more effective to combine techniques in order to drive users mad. For example, creating a fake incoming call that the user can't ever possibly answer!

Those who are dedicated enough should be able to work out how to adapt the following so that the notification looks plausible enough, but for those with truly ~~stupid~~ gullible users should find that the basics are often enough

```
xauth merge /home/$VICTIM/.Xauthority
ps aux | grep X
# Should give something like
# tty7     Ss+  May11 4828:54 /usr/bin/X :0 vt7 -nr -nolisten tcp -auth /var/run/xauth/A:0-
OpOMeb

export DISPLAY=:0
cd /usr/share/skype/sounds

# The sound is 3.3 seconds long, so we want to make it play a few times

kdialog --title "Incoming Call" --passivepopup "Human Resources" 10 && mplayer \
CallRingingIn.wav CallRingingIn.wav CallRingingIn.wav
```

The user will then hear the Skype call sound, and see a passive popup notifying them of a new call. It's possible to do more using QT but that requires far more dedication.

Similarly, the ability to combine techniques means that you can make it very embarrassing for users without needing to find a website that actually makes sound! Whilst it's easy enough to find a YouTube video of something particularly offensive to the ears (We've mentioned him a few times now!), the downside is that users can easily stop that by closing the browser. Combining techniques makes it a little harder, especially if you combine the visual and audible effects sensibly

```
xauth merge /home/$VICTIM/.Xauthority
ps aux | grep X
export DISPLAY=:0

cd /root
mplayer Justin_Bieber-baby.mp3 & konqueror “http://www.bieberfever.com”
```

The ability to sensibly combine techniques is what's needed to effectively drive people insane.

Conclusion

Hopefully this book has given you a few ideas for winding users up. Whilst more could have been included, the best part of a really good prank is the meticulous planning that is put in by the best practitioners. If nothing else, this book should have given you an insight into the many, many opportunities available to you.

Always make a backup of any files you are changing, and avoid playing pranks on anyone that may overreact and fire you (or refuse to cook your dinner!) unless you're reasonably sure that they'll find it funny. If in doubt ~~do it anyway~~ err on the side of caution.

About The Author

I work as Virya Technologies as an IT Manager & Linux specialist. Unfortunately, I don't get opportunity to play pranks on users (otherwise known as customers) because that's often considered to be bad for business! Unfortunately for my colleagues (and wife), however, I does have networks that I can use to play pranks which gives me a chance to exercise my sense of humour (all in the name of testing security).

I've worked in a variety of IT roles, and have had ample opportunity to test security on various networks even when I've been lacking in opportunities to play tricks!

Some people don't appreciate my particular flavour of humour, so apologies if you are one of them (thanks for buying my book though!). As a result, most of the writing I do is on the more serious and technical side.

You can see more of my work at the locations below

http://www.viryatechnologies.com
http://www.bentasker.co.uk

You can also buy my other book "Linux for Business People" from Amazon.