Building the Perfect Backtrack 4 USB Thumb Drive

This how-to will show you a method for building a USB thumb drive with the following features:

- Persistent Changes – Files saved and changes made will be kept across reboots.
- Nessus and NessusClient installed – Everybody needs Nessus
- Encryption configured (Note: This is not whole drive encryption)

We will also tweak a few things and make some interesting changes.

Table of contents:

Tools and Supplies
Partition the USB thumbdrive
Make a bootable Backtrack 4 USB thumbdrive
Persistent Changes
Install Nessus
Configure Encryption
Tweak a few things

Tools and Supplies

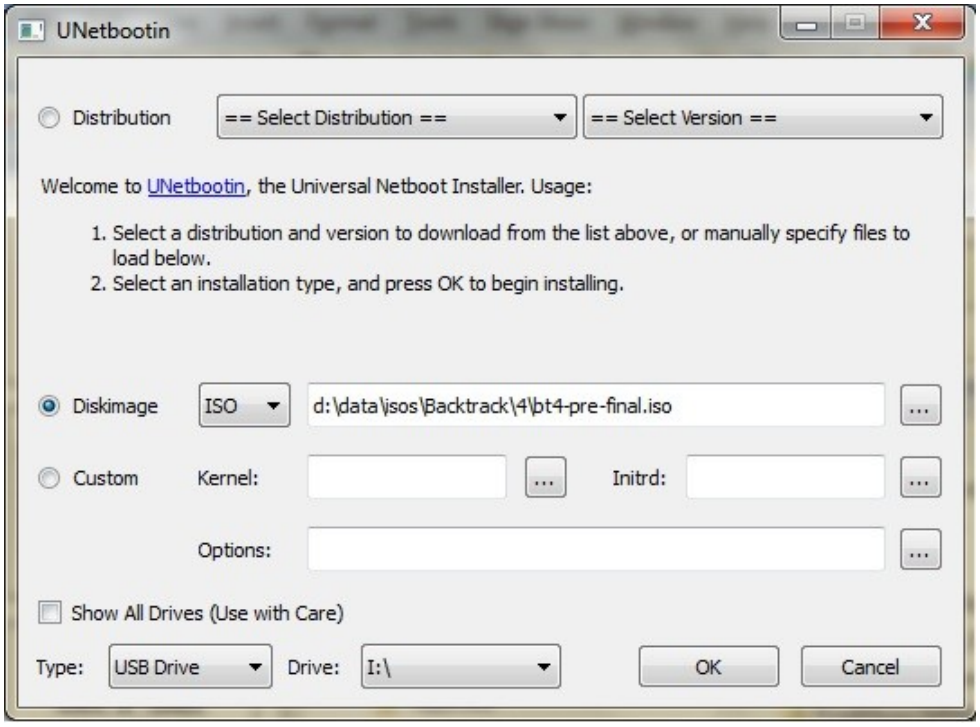
1. A USB thumbdrive – minimum capacity 4GB
2. A Backtrack 3 CDROM, Backtrack 4 DVD or an additional USB thumbdrive (minimum 2GB) – Used to partition the thumbdrive.
3. Optional: UNetbootin – A tool to transfer an iso image to a USB drive.

Let's get started!

Let's grab a copy of the Backtrack 4 Pre Release ISO.

Description: DVD Image
Name:: bt4-pre-final.iso
Size: 1390 MB
MD5: b0485da6194d75b30cda282ceb629654
Download: Click here

Now that we have the goods in hand, we can get to cooking. This tutorial is based on booting Backtrack 4 first. This means that you need some form of bootable Backtrack 4 media. This can be a virtual machine, DVD, or USB drive. Use your favorite method of creating a DVD or USB drive or you can use UNetBootin to create the thumb drive. Below is a screenshot of using UnetBootin to install Backtrack 4 on a USB drive.



It is a simple as selecting the image we want to write to the USB drive, the drive to write it to, and then clicking the 'OK' button. Warning: Make sure you pick the correct destination drive. You don't want to shoot yourself in the foot. 

Partition the USB thumbdrive

The first step is to boot up Backtrack 4. With the release of Backtrack 4 Final, a 4 GB drive is required if we are going to enable persistence. For Backtrack 3 and Backtrack 4 Beta, we could get away with a 2GB drive. We will also need to figure out which drive is our target drive. The following command will show the drives available and you can determine from that which is the new USB drive:

dmesg | egrep hd.\|sd.

We need to partition and format the drive as follows:

1. The first partition needs to be a primary partition of at least 1.5 GB and set to type vfat. Also remember to make this partition active when you are creating it. Otherwise you might have some boot problems.
2. The second Partition can be the rest of the thumb drive.

Below are the steps to take to get the drive partitioned and formatted. These steps are taken from this video on Offensive Security website. A '# blah blah' indicates a comment and is not part of the command and user typed commands are **bolded**. One note, we will need to delete any existing partitions on the drive.

```
fdisk /dev/sda # use the appropriate drive letter for your system

# delete existing partitions. There may be more than one.

Command (m for help): d
Partition number (1-4): 1

# create the first partition

Command (m for help): n
Command action
e   extended
```

```
p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-522, default 1): <enter>
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-522, default 522): +1500M

#create the second partition

Command (m for help): n
Command action
e   extended
p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (193-522, default 193): <enter>
Using default value 193
Last cylinder, +cylinders or +size{K,M,G} (193-522, default 522): <enter>
Using default value 522

# Setting the partition type for the first partition to vfat/fat32

Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): b
Changed system type of partition 1 to b (W95 FAT32)

# Setting the partition type for the second partition to Linux

Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 83

# Setting the first partition active

Command (m for help): a
Partition number (1-4): 1

Command (m for help): w

# now it is time to format the partitions

mkfs.vfat /dev/sdb1
mkfs.ext3 -b 4096 -L casper-rw /dev/sdb2
```

Two things to notice above in the format commands; 1) we are using ext3 instead of ext2 and 2) you must include the -L casper-rw portion of the command. Being able to use ext3 is great because of journaling. The -L casper-rw option helps us get around the problem we had where we had to enter the partition name in order to get persistence working. As you will see, that is no longer necessary. WooHoo!

So go ahead and partition and format the drive according the layout above.

Make it a bootable Backtrack 4 USB thumb drive

In the previous version of this how-to, we used UNetBootin to copy the ISO to the thumb drive and make it bootable. That required us to boot back to windows and then back again to Backtrack. We are changing to doing everything from Backtrack now. These steps are also taken from the Offensive Security video mentioned above.

The steps are basically:

1. Mount the first partition.
2. Copy the Backtrack files to it.
3. Install grub.

Following are the commands to execute. Again, '#' denote comments and user typed commands are in **bolded**.

```
# mount the first partition, sda1 in my case.

mkdir /mnt/sda1
mount /dev/sda1 /mnt/sda1

# copy the files, you will need to find where the ISO is mounted on your system.

cd /mnt/sda1
rsync -r /media/cdrom0/* .

# install grub

grub-install --no-floppy --root-directory=/mnt/sda1 /dev/sda
```

That's it. We now have a bootable Backtrack 4 USB thumb drive. Now on to setting up persistent changes.

Persistent Changes

This is done much differently and more easily than it was in Backtrack 4 Beta or Backtrack 3. First of all, for basic persistence, we don't have to do anything at all. There is already a menu option that takes care of it for us. Unfortunately, it is only for console mode so we need to make a couple changes.

We want to do the following things:

1. Change the default boot selection to persistent.
2. Set the resolution for our gui.

To do so, do the following. Again, '#' …comment….user typed…blah blah.

```
cd /mnt/sda1/boot/grub

vi menu.lst

# change the default line below to 'default 4' and append 'vga=0×317' (that's a zero) to the kernel line to set
the resolution to 1024×768

        # By default, boot the first entry.
        default 4
        .
        .
        .
        title               Start Persistent Live CD
        kernel          /boot/vmlinuz BOOT=casper boot=casper persistent rw quiet vga=0×317
        initrd           /boot/initrd.gz

:wq
```

Here is my entire menu.lst file for reference.

```
# By default, boot the first entry.
default 4

# Boot automatically after 30 secs.
timeout 30

splashimage=/boot/grub/bt4.xpm.gz

title                Start BackTrack FrameBuffer (1024×768)
kernel                /boot/vmlinuz BOOT=casper boot=casper nopersistent rw quiet vga=0×317
initrd                /boot/initrd.gz
```

```
title                Start BackTrack FrameBuffer (800x600)
kernel               /boot/vmlinuz BOOT=casper boot=casper nopersistent rw quiet vga=0x314
initrd               /boot/initrd800.gz

title                Start BackTrack Forensics (no swap)
kernel               /boot/vmlinuz BOOT=casper boot=casper nopersistent rw vga=0x317
initrd               /boot/initrdfr.gz

title                Start BackTrack in Safe Graphical Mode
kernel               /boot/vmlinuz BOOT=casper boot=casper xforcevesa rw quiet
initrd               /boot/initrd.gz

title                Start Persistent Live CD
kernel               /boot/vmlinuz BOOT=casper boot=casper persistent rw quiet vga=0x317

initrd               /boot/initrd.gz

title                Start BackTrack in Text Mode
kernel               /boot/vmlinuz BOOT=casper boot=casper nopersistent textonly rw quiet
initrd               /boot/initrd.gz

title                Start BackTrack Graphical Mode from RAM
kernel               /boot/vmlinuz BOOT=casper boot=casper toram nopersistent rw quiet
initrd               /boot/initrd.gz

title                Memory Test
kernel               /boot/memtest86+.bin

title                Boot the First Hard Disk
root                (hd0)
chainloader +1
```

Reboot and either select “Start Persistent Live CD” or just wait since we set it to auto-boot to persistent mode. To test it, create a file and reboot again. If your file is still there, everything is golden.

Install Nessus

Now that our changes are saved from boot to boot, we can install things and they won’t disappear on us

Download the Ubuntu Nessus package from nessus.org. The 32-bit 8.10 version worked fine for me. We used to have to install a separate client package, but no longer. The client is now web-based and included in the Nessus package.

Again, with Backtrack 4 things are little easier. To install the Nessus server, simply execute the following command to install the package.

```
dpkg --install Nessus-4.2.-ubuntu810_i386.deb
```

Finally it's time to configure Nessus. Another step that is no longer necessary is the creation of certificates for authentication, so all we really need to do is add our user.

```
# add user

/opt/nessus/sbin/nessus-adduser

Login :Me
Authentication (pass/cert) : [pass]<enter>
Login password :
Login password (again) :
Do you want this user to be a Nessus ‘admin’ user ? (can upload plugins, etc…) (y/n) [n]:y
User rules
----
nessusd has a rules system which allows you to restrict the hosts
that Me has the right to test. For instance, you may want
him to be able to scan his own host only.
```

```
Please see the nessus-adduser manual for the rules syntax

Enter the rules for this user, and enter a BLANK LINE once you are done :
(the user can have an empty rules set)

Login             : Me
Password          : ***********
This user will have 'admin' privileges within the Nessus server
Rules             :
Is that ok ? (y/n) [y]y
User added

We want to disable Nessus starting at boot. We are going to do some things a little later than
require that Nessus not be running at boot.

/usr/sbin/update-rc.d -f nessusd remove
```

This command does not remove the Nessus start scripts. It only removes the links that cause Nessus to start at boot time.

The next thing we need to do is register our installation so we can get the plugin feed. You need to go here and request a key. That is a link to the free feed for home use. Use appropriately.

Once you have your key. Execute the following to update your plugins. Please note that there are two dashes before register in the nessus-fetch line below. They can display as one sometimes.

```
/opt/nessus/bin/nessus-fetch --register [your feed code here]
```

When that is done, and it is going to take a few minutes, you are ready to start the server and client. Be aware that with version 4.x, while the command to start returns quickly, the actual starting of the service may take a minute or two. In many cases, I have had to reboot after the initial install before Nessus started working. You can use 'netstat -napt' to check that the server is listening on port 8834. Yup, this is different too. We used to look for port 1241.

```
/etc/init.d/nessusd start
```

Woohoo, time to find those vulnerabilities.

Configure Encryption

Before we configure encryption, we need to go ahead and update the system. We used to be able to wait to do this, but the amount of packages is now enough that we run out of space if we wait until after creating the Truecrypt volume.

First execute the following:

```
apt-get update
```

This is update the software repository information. Next, execute the this command:

```
apt-get upgrade
```

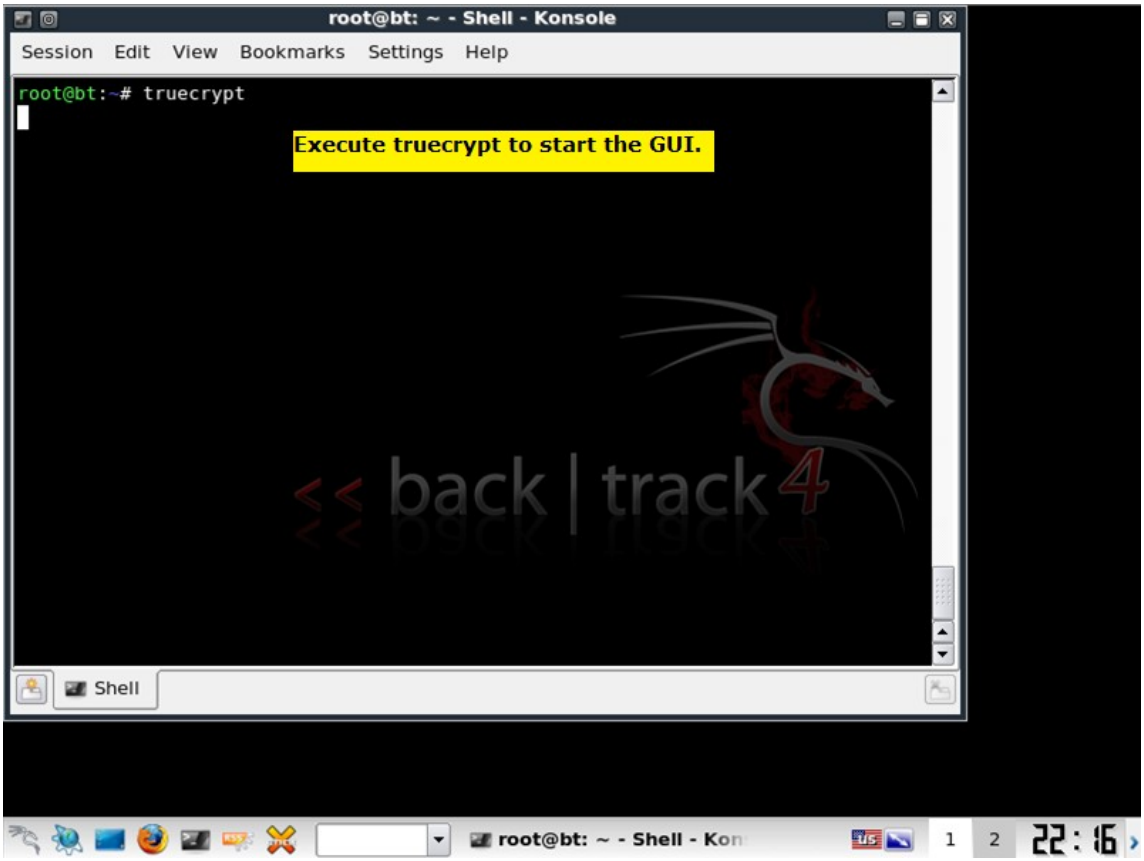
The system will determine if there is anything that needs to be updated and then prompt you to continue. Individual packages can be updated by including the package name after upgrade.

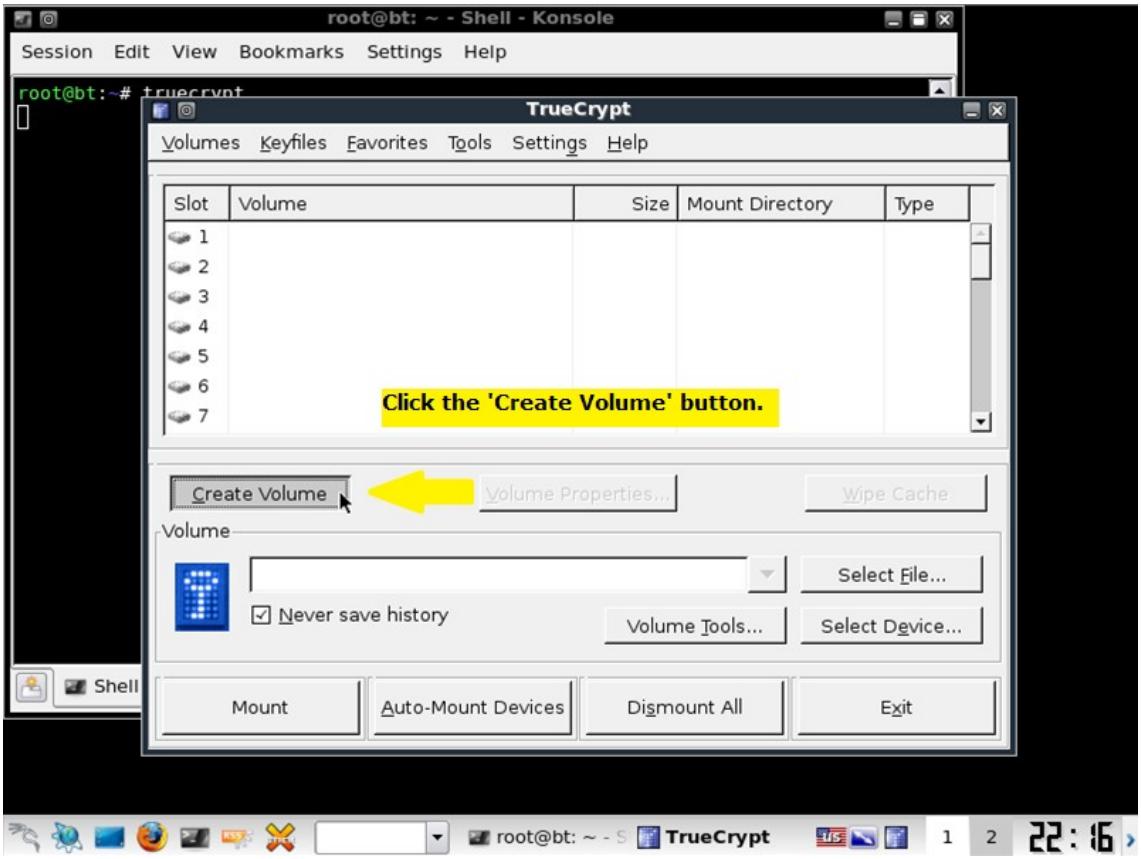
Finally, execute the following to clean up the downloaded packages and make room for the Truecrypt volume.

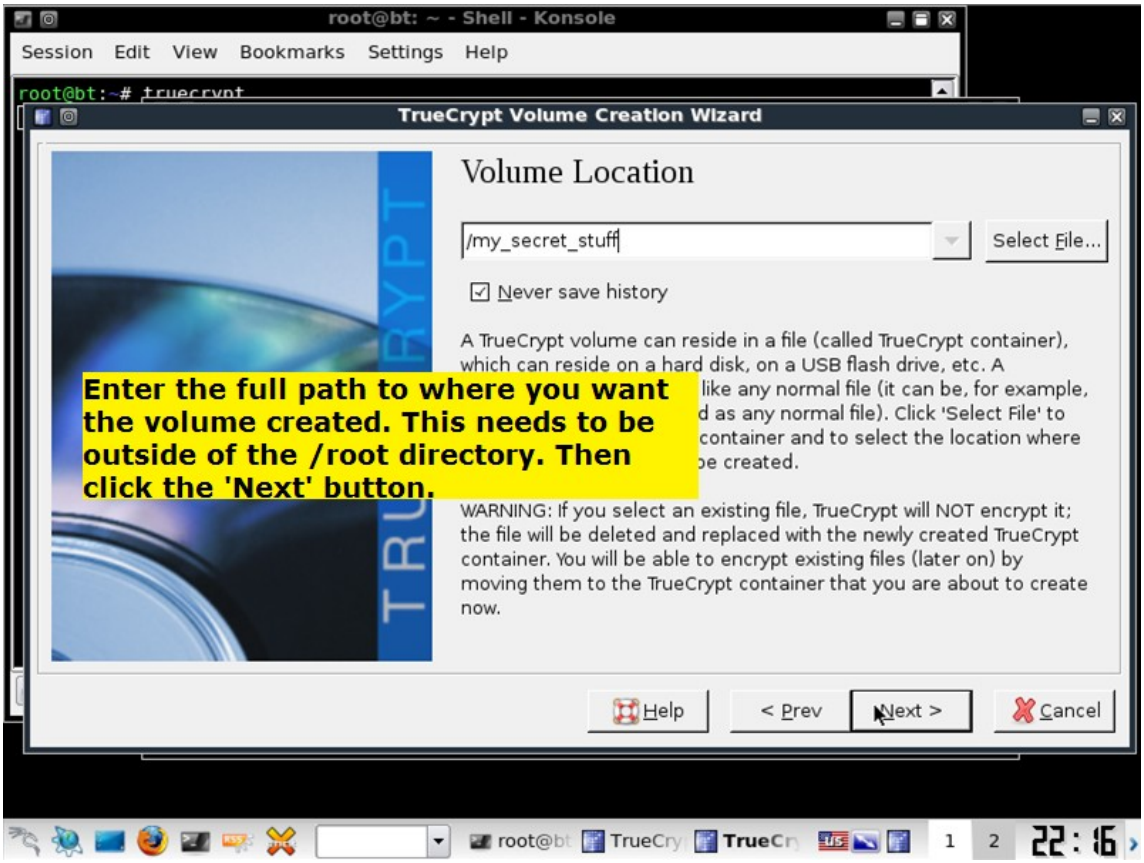
```
apt-get clean
```

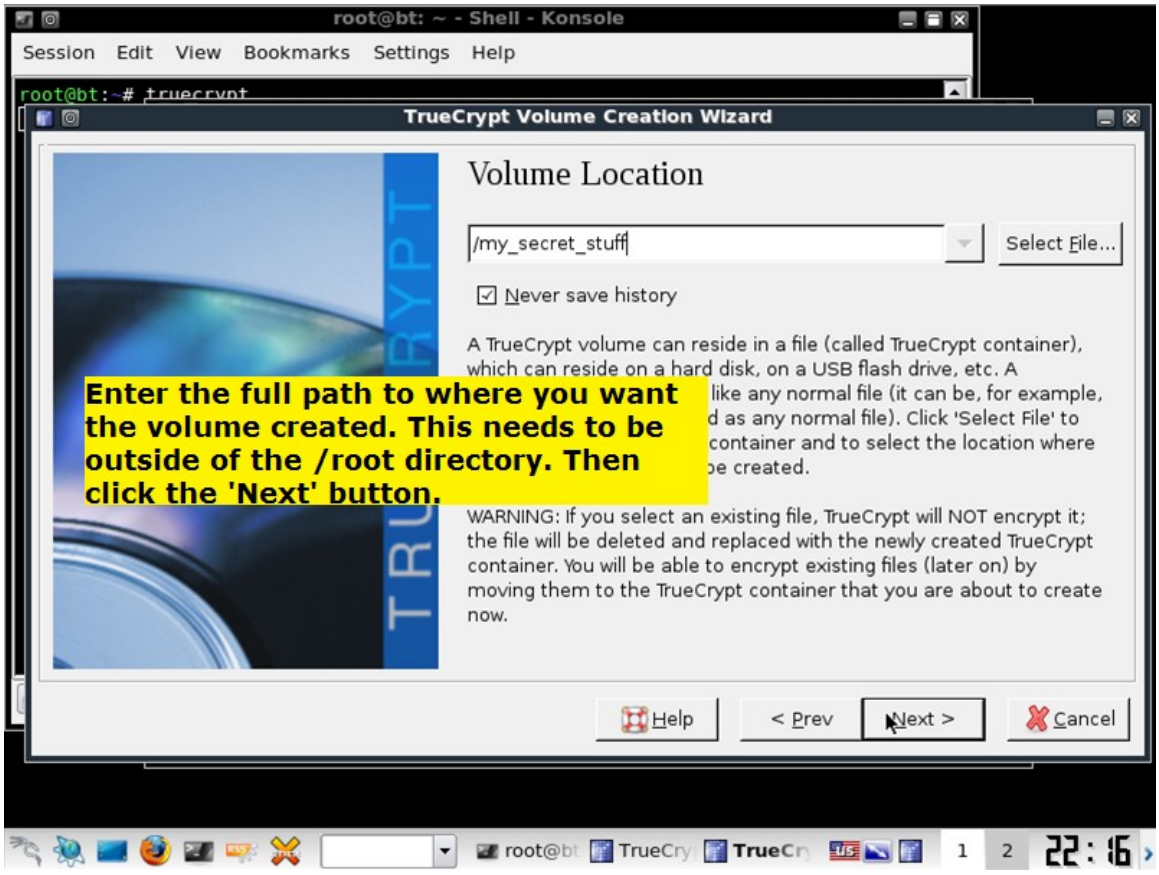
Since we are using this tool to poke at peoples networks and systems, with permission of course, it is very important that the information we find be protected. To do this, we are going to setup an encrypted volume that will eventually become our home directory.

This can be done with the gui or via command line. We will be using the gui because we need to be able to format the volume with ext3 and, as yet, I have not been able to figure out how to do that via the command line on linux. Click on the images to see a larger version.

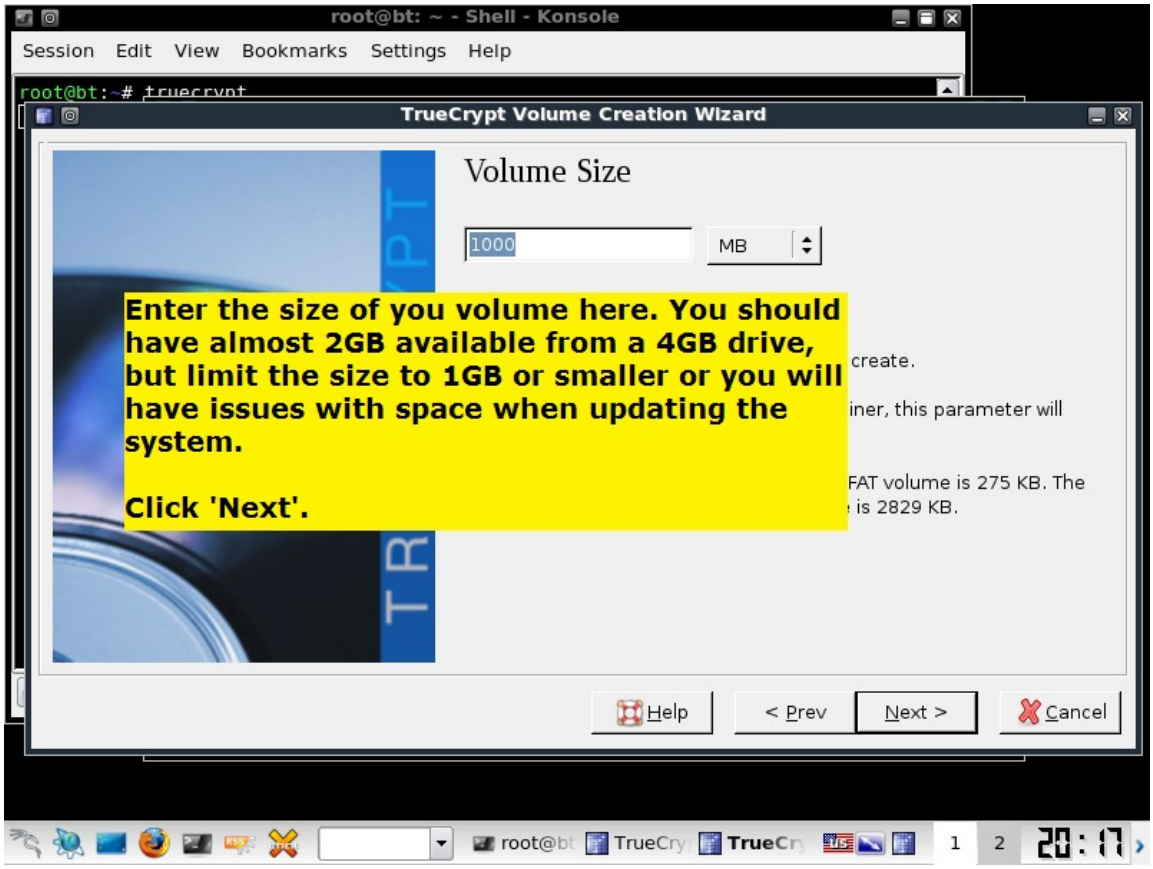


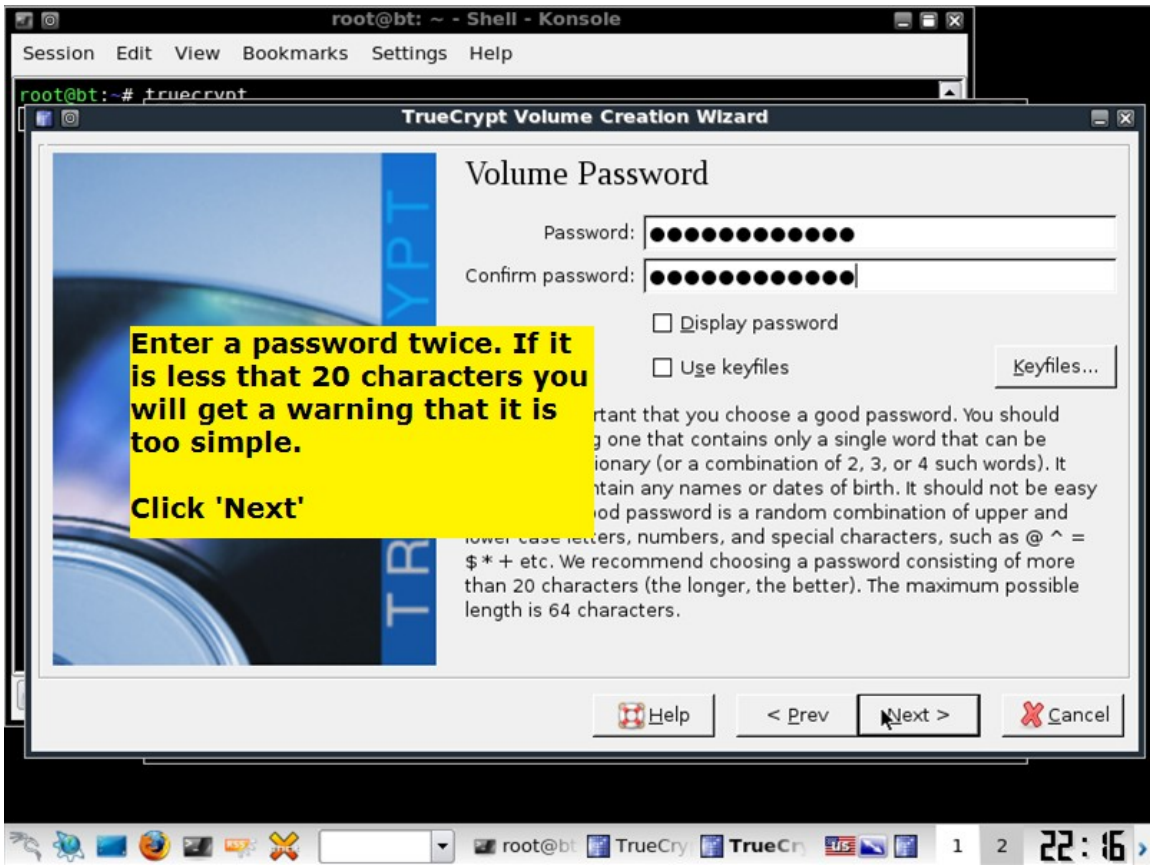

root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
root@bt:~# truecrypt
TrueCrypt
Volumes Keyfiles Favorites Tools Settings Help
Slot Volume Size Mount Directory Type
1
2
3
4
5
6
7
Click the 'Create Volume' button.
Create Volume
Volume Properties...
Wipe Cache
Volume
Select File...
Never save history
Volume Tools...
Select Device...
Mount
Auto-Mount Devices
Dismount All
Exit
Shell
root@bt: ~ - S
TrueCrypt
1
2
22:16

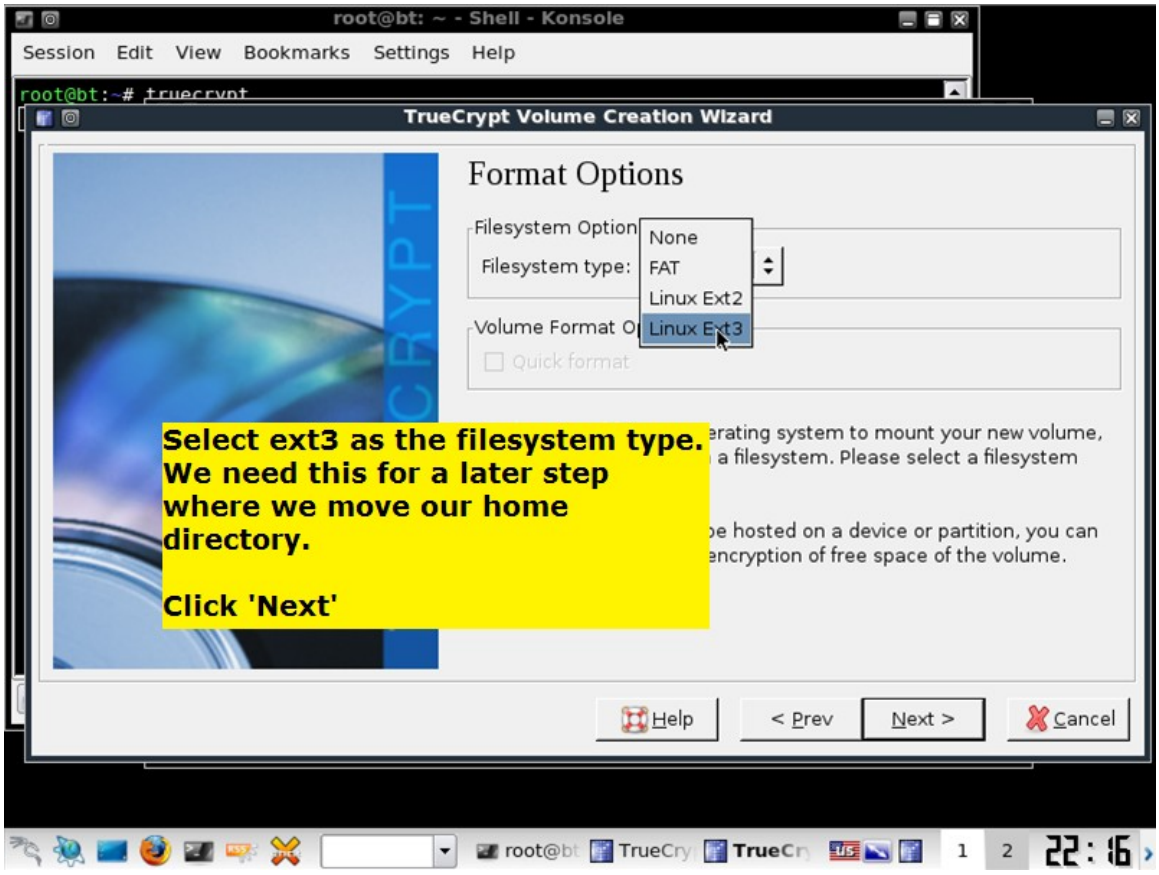

root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
root@bt:~# truecrypt
TrueCrypt Volume Creation Wizard
Volume Location
/my_secret_stuff
Select File...
Never save history
A TrueCrypt volume can reside in a file (called TrueCrypt container), which can reside on a hard disk, on a USB flash drive, etc. A
like any normal file (it can be, for example,
d as any normal file). Click 'Select File' to
container and to select the location where
e created.
Enter the full path to where you want the volume created. This needs to be outside of the /root directory. Then click the 'Next' button.
WARNING: If you select an existing file, TrueCrypt will NOT encrypt it; the file will be deleted and replaced with the newly created TrueCrypt container. You will be able to encrypt existing files (later on) by moving them to the TrueCrypt container that you are about to create now.
Help
< Prev
Next >
Cancel
root@bt
TrueCry
TrueCr
1
2
22:16

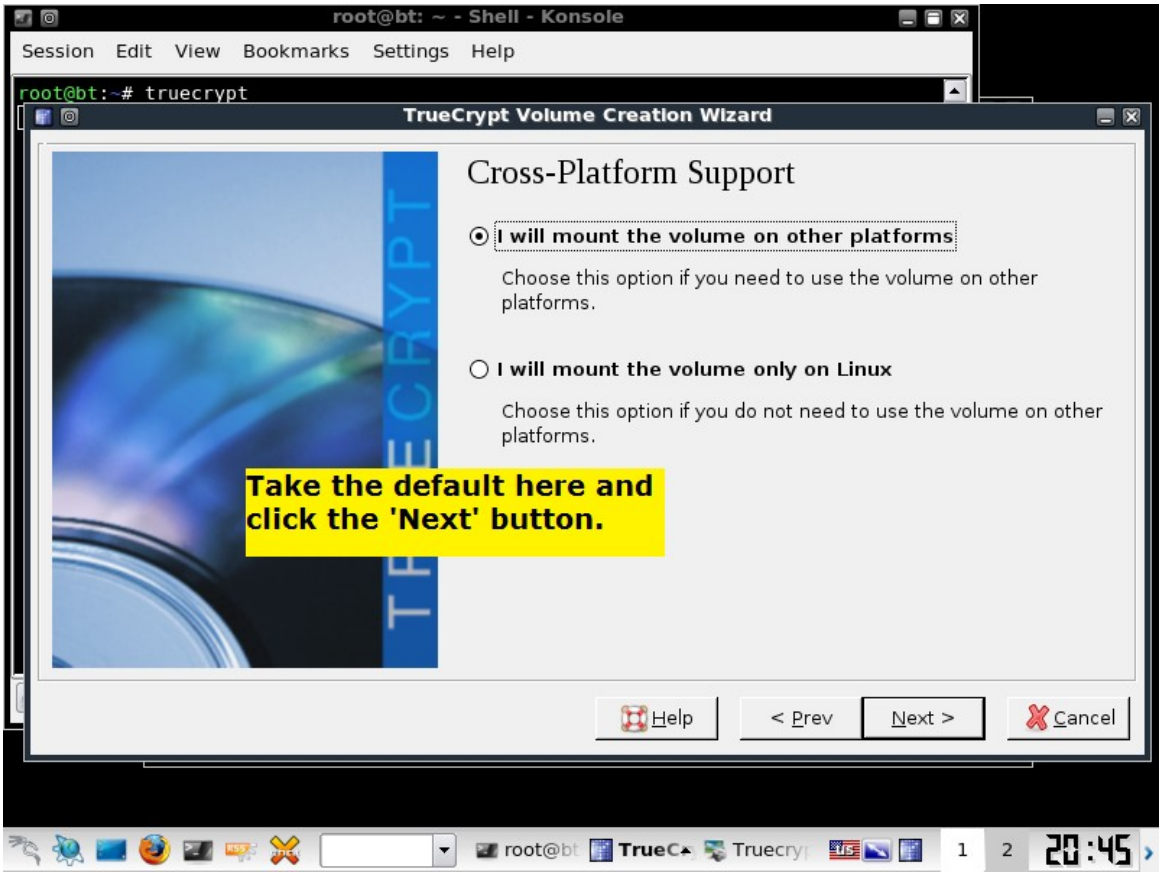

root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
root@bt:~# truecrypt
TrueCrypt Volume Creation Wizard
Volume Location
/my_secret_stuff
Select File...
Never save history
A TrueCrypt volume can reside in a file (called TrueCrypt container), which can reside on a hard disk, on a USB flash drive, etc. A
like any normal file (it can be, for example,
d as any normal file). Click 'Select File' to
container and to select the location where
e created.
Enter the full path to where you want the volume created. This needs to be outside of the /root directory. Then click the 'Next' button.
WARNING: If you select an existing file, TrueCrypt will NOT encrypt it; the file will be deleted and replaced with the newly created TrueCrypt container. You will be able to encrypt existing files (later on) by moving them to the TrueCrypt container that you are about to create now.
Help
< Prev
Next >
Cancel

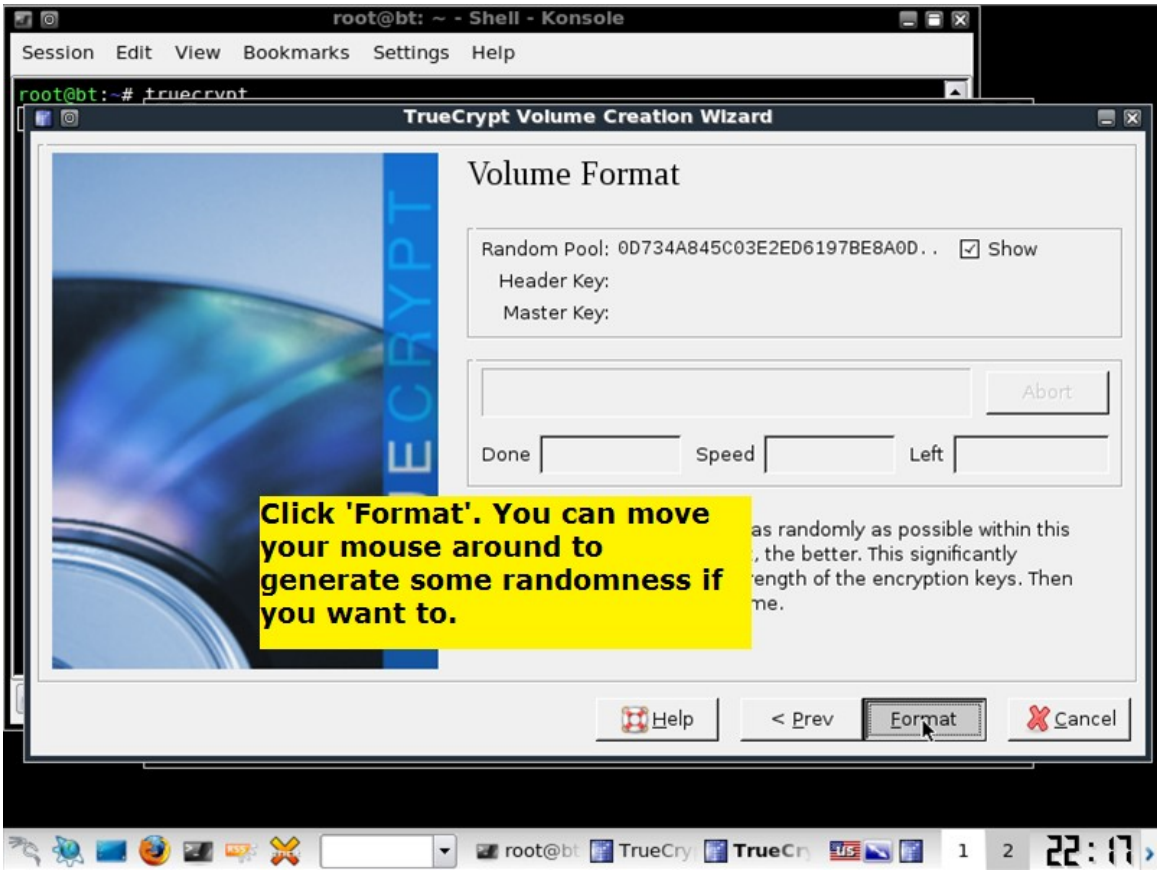

root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
root@bt:~# truecrypt
TrueCrypt Volume Creation Wizard
Encryption Options
Encryption Algorithm
AES
Test
FIPS-approved cipher (Rijndael, published in 1998) that may be used by U.S. government departments and agencies to protect classified information up to the Top Secret level. 256-bit key, 128-bit block, 14 rounds (AES-256). Mode of operation is XTS.
Take the defaults here and click the 'Next' button.
More information on AES
Hash Algorithm
RIPEMD-160
Information on hash algorithms
Help
< Prev
Next >
Cancel


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
root@bt:~# truecrypt
TrueCrypt Volume Creation Wizard
Volume Size
1000
MB
Enter the size of you volume here. You should have almost 2GB available from a 4GB drive, but limit the size to 1GB or smaller or you will have issues with space when updating the system.
Click 'Next'.
create.
iner, this parameter will
FAT volume is 275 KB. The
is 2829 KB.
Help
< Prev
Next >
Cancel


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
root@bt:~# truecrypt
TrueCrypt Volume Creation Wizard
Volume Password
Password:
Confirm password:
Display password
Use keyfiles
Keyfiles...
Enter a password twice. If it is less that 20 characters you will get a warning that it is too simple.
Click 'Next'
rtant that you choose a good password. You should
one that contains only a single word that can be
ionary (or a combination of 2, 3, or 4 such words). It
ntain any names or dates of birth. It should not be easy
od password is a random combination of upper and
lower case letters, numbers, and special characters, such as @ ^ = $ * + etc. We recommend choosing a password consisting of more than 20 characters (the longer, the better). The maximum possible length is 64 characters.
Help
< Prev
Next >
Cancel


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
root@bt:~# truecrypt
TrueCrypt Volume Creation Wizard
Format Options
Filesystem Option
Filesystem type:
None
FAT
Linux Ext2
Linux Ext3
Volume Format O
Quick format
Select ext3 as the filesystem type. We need this for a later step where we move our home directory.
Click 'Next'
Help
< Prev
Next >
Cancel


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
root@bt:~# truecrypt
TrueCrypt Volume Creation Wizard
Cross-Platform Support
I will mount the volume on other platforms
Choose this option if you need to use the volume on other platforms.
I will mount the volume only on Linux
Choose this option if you do not need to use the volume on other platforms.
Take the default here and click the 'Next' button.
Help
< Prev
Next >
Cancel



You will get a message that the volume was successful created. Click on the 'OK' button, then exit the Truecrypt gui, both the 'Create Volume' windows and the main windows. We want to be back at the command prompt at this point.

If you want to test the your filesystem, execute the following, note the -k " is two single quotes, not a double quote:

```
truecrypt -t -k " --protect-hidden=no /my_secret_stuff /media/truecrypt1
mount
cd /media/truecrypt1
df .
```

This will show that the volume is mounted and the amount of disk space you have left. Our next step is to have this volume mounted when we log in. We do this by editing the root user's .profile file. Add the truecrypt command above to root's .profile so it looks like this:

```
# ~/.profile: executed by Bourne-compatible login shells.
if [ "$BASH" ]; then
  if [ -f ~/.bashrc ]; then
    . ~/.bashrc
  fi
fi

truecrypt -t -k '' --protect-hidden=no /my_secret_stuff /media/truecrypt1

mesg n
```

The next time you reboot you will be asked for the password for the volume and it will be mounted for you.

Now it is time to tweak a few tings

Tweak a few things

The first thing we are going to do is go ahead and configure networking to start at boot time. It's convenient and easy to disable if we need to. All we have to do is execute the following command.

```
/usr/sbin/update-rc.d networking defaults
```

This next bit is interesting and I was surprised it worked. We are going to reset the root user's home directory during the login process to the mounted truecrypt volume. This will ensure that anything written to the home directory will be encrypted. The following commands will set this up for us:

```
cd /media/truecrypt1

rsync -r -links /root/ .

# add the bold lines below

vi /root/.profile

  # ~/.profile: executed by Bourne-compatible login shells.
  if [ "$BASH" ]; then
    if [ -f ~/.bashrc ]; then
      . ~/.bashrc
    fi
  fi

  truecrypt -t -k '' --protect-hidden=no /my_secret_stuff /media/truecrypt1

  export HOME=/media/truecrypt1
  export HISTFILE=/media/truecrypt1/.bash_history

  cd

  mesg n

:wq
```

The next time you reboot, when you are finally in the system, your home directory will be /media/truecrypt1.

There is one last thing we want to do. We want to change nessus to log to the encrypted volume. This is very easy. The file that controls this is /opt/nessus/etc/nessus/nessusd.conf. We need to create a place for the log files to go. So execute the following

```
cd /media/truecrypt1

mkdir -p nessus/logs

Once you have done that, edit the /opt/nessus/etc/nessus/nessusd.conf file and change this:

.
.
.
# Log file :
logfile = /opt/nessus/var/nessus/logs/nessusd.messages

# Shall we log every details of the attack ? (disk intensive)
log_whole_attack = no

# Dump file for debugging output
dumpfile = /opt/nessus/var/nessus/logs/nessusd.dump
.
.
.
```

to this:

```
.
.
.
# Log file :
logfile = /media/truecrypt1/nessus/logs/nessusd.messages
```

```
# Shall we log every details of the attack ? (disk intensive)
log_whole_attack = no

# Dump file for debugging output
dumpfile = /media/truecrypt1/nessus/logs/nessusd.dump
.
.
.
```

That's it. You are all done now. Go forth and have fun.

Please let me know of any corrections or changes that should be made. You can leave a comment or send me a note at kriggins@infosecramblings.com.

Kevin Riggins, CISSP, CCNA
Website: http://www.infosecramblings.com
Twitter: kriggins
LinkedIn: http://www.linkedin.com/in/kevinriggins