Revealer Toolkit – User Guide

Jose Navarro

March 8, 2009

Contents

Chapter 1

License

1.1 License of this document

1.2 License of the Revealer Toolkit

For more information, please visit http://www.gnu.org/licenses/old–licenses/gpl–2.0.txt

Chapter 2

Acknowledgments

- INCIDE (Investigacion Digital S.L., www.incide.es) where developers and testers work
- Manu Ginés aka xkulio creator of the original Chanchullos Revealer
- Generalitat de Catalunya for partial funding of the project

Chapter 3

Introduction

the Revealer Toolkit is a framework and simple scripts for computer forensics. It uses Brian Carrier's The Sleuth Kit as the backbone, as well as other free tools.

The aim of the Revealer Toolkit is to automate rutinary tasks and to manage sources and results from another perspective than the usual forensic frameworks.

RVT is developed and actively tested by computer forensic investigators working at INCIDE, spanish company sited at the beautiful city of Barcelona (see www.incide.es for more details)

project state: we are using RVT actually in our production servers, although do not expect an easy and clean software yet or a quick installation. For any questions, help or comments, please, drop an email.

You can find additional information, packages and all the source code at http://code.google.com/p/revealertoolkit

Chapter 4

Installation

This is and example installation of the Revealer Toolkit Shell on a Debian *lenny* 5.00. Also covers the creation of a valid folder structure and a sample image.

4.1 Operating system

Download last Debian stable version. In this example, Debian 5.00 *lenny* is used.

Some additional packages are required to be installed with *apt-get*:

```
apt-get install openssh-server
apt-get install unzip
apt-get install sudo vim
apt-get install sleuthkit binutils
apt-get install dosfstools

groupadd -g 1010 forensics
addgroup analyst forensics

mkdir /media/morgue
chgrp forensics morgue
chmod g+sw morgue

mkdir /media/datos
chgrp forensics datos
chmod g+sw datos
```

```
chmod a+r /dev/loop*
and add the last line to the  \emph{/etc/init.d/bootmisc.sh}  file (before the last
```

as root, add this line to the sudoers file (with *visudo* command):

```
Defaults:%forensics !authenticate

%forensics ALL=(root) /bin/mount, (root) /bin/umount, (root) /sbin/losetup
```

4.2 Software

```
wget revealertoolkit.googlecode.com/files/RVT_v0.1.zip
unzip RVT_v0.1.zip
chmod a+x RVT/RVT.pl
```

as root:

```
ln -s /home/analyst/RVT/RVT.pl /usr/bin/RVT.pl
```

4.3 Folder structure

as user:

```
mkdir /media/morgue/imagenes
mkdir /media/datos/imagenes

mkdir /media/morgue/imagenes/100101-ghost
mkdir /media/morgue/100101-ghost
mkdir /media/morgue/100101-ghost/100101-01-1
mkdir /media/morgue/100101-ghost/100101-01-1/mnt
mkdir /media/morgue/100101-ghost/100101-01-1/output
```

4.4 Sample image creation

as user:

```
cd /media/morgue/imagenes/100101-ghost/
dd if=/dev/zero bs=1024 count=10240 > 100101-01-1.dd
```

as root:

```
fdisk /media/morgue/imagenes/100101-ghost/100101-01-1.dd

in fdisk:

x (additional functions)

c (cylinders)
1024
s (sectors)
10
h (heads)
2

r (main menu)
n (new partition)
p
1
1   (first cylinder)
512 (half of the disk)
n (new partition)
p
2
513 (half of the disk)
1024 (last cylinder)
w (save and exit)
```

and now, let's put some info inside, as root:

```
echo "losetup assign 100101" | RVT.pl -b
echo "losetup list" | RVT.pl -b
   (the last command reveals the loop devices created)
mkfs.vfat /dev/loop0
mkfs.vfat -F 32 /dev/loop1

mkdir /media/aux1
mount /dev/loop0 /media/aux1
cp /home/analyst/RVT/RVT.pl /media/aux1
umount /media/aux1
mount /dev/loop1 /media/aux1
cp /home/analyst/RVT/RVT.pl /media/aux1
sync
```

```
rm /media/aux1/RVT.pl
echo "my email address is myemail@revealertoolkit.com" > /media/aux1/textfile.txt
umount /media/aux1

echo "losetup delete 100101" | RVT.pl -b
```

4.5 First test

Just for testing, some first commands can be executed. As analyst, execute the RVT Shell:

```
\$ RVT.pl
```

After the preliminar scanning, RVT Shell will offer you a prompt:

```
RVT >
```

Now, execute these commands:

```
set level 100101-01-1
script strings generate
script timelines generate
script webmail detection
script software detection
script search quickcount emails
script search quickcount accounts
quit
```

You can check that the directory */media/morgue/100101-ghost/100101-01-1/output* has been populated with results.

The same can be achieved creating a file with the commands and piping it to the RVT Shell using the *-b* argument:

```
cat preforensics.rvt | RVT -b
```

Chapter 5

Folder structure

The morgue stores disk images and results of forensic analysis. Each morgue must have a strict folder structure.

5.1 Morgues

The Revealer Toolkit can handle more that one morgue. By default, two morgues are defined in RVT, mounted at */media/morgue* and */media/datos*.

These morgues can be managed modifying the RVT Shell internal code. See the RVT Developer Guide for more info.

5.2 Cases

Each forensic case is determined by a *case number* and a *case codename*, separated by a dash. For example, the example case created in the Chapter 4 is noted as *100101-ghost*.

Each case has a folder assigned in the morgue, under the folder *images*, where the disk images are stored.

Also, each case has a folder in the morgue, where all the forensic results are stored.

The folder structure, at the case level, for the example installation shown in chapter 4, will be:

```
/media/morgue/100101-ghost
```

```
/media/morgue/imagenes/100101-ghost
/media/morgue/imagenes/100101-ghost/100101-01-1.dd
```

where *100101-01-1.dd* is the dd image of a disk.

5.3 Devices, disks and partitions

Under the Revealer Toolkit, information sources are organized with:

- Devices: each case has a number of devices: computers, cell phones, digital cameras, ... They are numbered sequencially from 01 to 99.

  A device is noted as *casenumber-devicenumber*, for example, *100101-01* for the case 100101 and device 01.

- Disks: each device has a number of disks: hard disks, CD's, memory cards, ... They are numbered sequencially from 1 to 9

  A disk is noted as *case-device-disknumber*, for example, *100101-01-1*, for the disk 1

- Partitions: each disk can have several partitions, numbered from 01 to 99. The numeration used by the Sleuthkit command *mmls* is used.

  A partition is noted as *case-device-disk-ppartition*, for example, 100101-01-1-p02 for partition 02.

Under each case folder, a folder must exist for every disk to be analyzed.

This folder structure, at a disk level, for the example installation shown in chapter 4, will be:

```
/media/morgue/100101-ghost
/media/morgue/100101-ghost/100101-01-1

/media/morgue/imagenes/100101-ghost
/media/morgue/imagenes/100101-ghost/100101-01-1.dd
```

5.4 Forensic results

The Revealer Toolkit Shell manages and executes *script modules*, which performs forensic operations on the disk images and disk information. The

results of these script modules are stored in the corresponding disk folder, under a folder named *output*.

The folder structure and file content stored here depends on each script module. See chapter 7 for further information.

Furthermore, under the disk folder other folder exists, named *mnt*, that contains the mounting points for the image partitions.

Then, the complete folder structure for the example shown in chapter 4 will be:

```
/media/morgue/100101-ghost
/media/morgue/100101-ghost/100101-01-1
/media/morgue/100101-ghost/100101-01-1/mnt
/media/morgue/100101-ghost/100101-01-1/output

/media/morgue/imagenes/100101-ghost
/media/morgue/imagenes/100101-ghost/100101-01-1.dd
```

Chapter 6

Command guide

The Revealer Toolkit Shell provide several commands used to (a) manage your forensic images and (b) execute forensic operations over them.

RVT Shell is a Perl script that, when executed, performs a scan of the morgue and of all the images stored into it. After that, a prompt is shown and commands can be introduced.

6.1 General interaction

- to execute one command, type it at the prompt, add the corresponding arguments, and type RETURN. Some commands return information on the screen, some, write information at the *output* folder of the corresponding disk at the morgue. Other, do both.
- type one command followed by *?* to obtain help about it
- type the TAB key to obtain a list of available commands

```
Welcome to Revealer Tools Shell (v0.1.0):

 RVT  >
        case
        images
        info
        losetup
        mount
        script
        set
        test
```

```
 RVT  >
```

6.2 Command: *case*

Case management.

case list

Gives a list of the cases stored in the morgue.

```
 RVT  > case list
Cases in the morgue:
        100101 'ghost':
                100101-01-1
```

6.3 Command: *images*

Image management.

images list

Gives a list of the disk images stored in the morgue.

```
 RVT  > images list
Images in the morgue:
        100101 'ghost':
                100101-01-1.dd
```

images partition info

Gives information about a partition.

```
 RVT  > images partition info 100101-01-1-p03

Info for partition  - 100101-01-1-p03:
```

```
Filesystem:     FAT12
Cluster size:   2048
Sector size:    512
Offset:         0000010240 sectors ( 5242880 bytes )
```

images partition table

Gives the partition table of an image.

```
RVT  > images partition table 100101-01-1

       03:     6 MB    Linux (0x83)
       02:     5 MB    Linux (0x83)
```

images scanall

RVT scans the morgue. No output given.

6.4 Command: *info*

Manages RVT Shell configuration information.

info list

Gives RVT Shell configuration information.

```
RVT  > info list

       List of morgues:
               /media/morgue
               /media/datos


       List of morgues of images:
               /media/morgue/imagenes
               /media/datos/imagenes
```

6.5 Command: *losetup*

Loop device management.

losetup assign

Assigns a loop device to each partition of a case.

```
 RVT  > losetup assign 100101

sudo losetup -f /media/morgue/imagenes/100101-ghost/100101-01-1.dd -o 5242880

sudo losetup -f /media/morgue/imagenes/100101-ghost/100101-01-1.dd -o 5120
```

losetup delete

Delete loop devices assigned to a case

```
 RVT  > losetup delete 100101
```

losetup list

List all loop devices assigned.

```
 RVT  > losetup list
Loop devices:
        loop1   100101-01-1.dd  5120
        loop0   100101-01-1.dd  5242880
```

losetup recheck

Updates loop device information from the operating system.

6.6 Command: *mount*

Mount points management.

mount assign

Mounts the partitions of a case at the path *morgue/100xxx-case/100xxx-device-disk/mnt/p0N*, where N is the partition number.

```
 RVT  > mount assign 100101

sudo mount /media/morgue/imagenes/100101-ghost/100101-01-1.dd /media/morgue/100101-g

sudo mount /media/morgue/imagenes/100101-ghost/100101-01-1.dd /media/morgue/100101-g
```

mount delete

Umounts partitions of a case.

```
 RVT  > mount delete 100101
```

mount list

List all mounted points.

```
 RVT  > mount list
Mounted partitions:
        100101-01-1.dd  loop=/dev/loop0 offset=5242880
        100101-01-1.dd  loop=/dev/loop1 offset=5120
```

mount recheck

Updates mount points information from the operating system.

6.7 Command: *script*

Modular scripts. Each module is explained in a separated chapter (see Chapter 7 for further information)

6.8 Command: *set*

Sets RVT Shell configuration information.

set level

Sets the work level to a specific case, device, disk or partition. When the work level is stablished, is notified at the prompt and there is no need of indicate it as argument at the commands.

```
 RVT  > images partition table 100101-01-1

        03:     6 MB    Linux (0x83)
        02:     5 MB    Linux (0x83)

 RVT  >
 RVT  > images partition table
```

```
I don't know what is this
```

```
 RVT  > set level 100101-01-1

 new format: disk
 RVT 100101-01-1 > images partition table

        03:     6 MB    Linux (0x83)
        02:     5 MB    Linux (0x83)

 RVT 100101-01-1 >
```

Chapter 7

Script modules

Scripts modules are forensic software components that perform specific forensic tasks on the stored images and information.

Each module has particular objectives, methods, arguments and results, the documentation of which is detailed in the following sections.

7.1 *script search*

7.2 *script software*

7.3 *script strings*

7.4 *script timelines*

7.5 *script webmail*