Autopsy and Sleuthkit, the Digital Forensics Toolkit

The Tracker Dog's Guide

Following a system compromise, the admin has to look for telltale signs and secure evidence – the admin becomes a forensic scientist. Sleuthkit and Autopsy can help with this difficult task using a practical Web interface to search for deleted files and discover traces of the intruders.

BY RALF SPENNEBERG

Sleuthkit searches Microsoft and Unix filesystems for deleted files and reconstructs the events leading up to an intrusion. Last month, we looked into the use of command line tools for this task [1]. The Autopsy Forensic Browser [2] is not only easier to use, it also provides more advanced functions: this web-based Sleuthkit front-end both facilitates and documents the process of forensic analysis.

New Cases

Just like in last month's issue, we will be basing our examples on the Forensic Challenge [4] filesystems. The tarball contains the individual partitions as described in Table 1.

Before starting an investigation with Autopsy and the Sleuthkit, forensic investigators first need to open a new case. To do so you can simply click on the *New Case* button at the bottom of the Autopsy welcome page. Doing so opens the input page for the new case (see Figure 2).

After you have filled out the fields and clicked on *New Case*, Autopsy will create the case directory (*/var/morgue/forensic_challenge/*) and the configuration file (*/var/morgue/forensic_challenge/case.aut)*, as well as adding the investigator. The tool displays the results on another webpage and prompts you to confirm by clicking on *OK*.

In the next window (Case Gallery, Figure 3), Autopsy presents a selectable list of cases; you can also access this page directly from the welcome page by clicking on the *Open Case* link. The *forensic_challenge* case is selected by default; after clicking on *OK* to confirm your selection, it is now time to add the computers you will be investigating to this case.

Hosts under the Magnifying Glass

To add a new host you must specify the name of the computer and can optionally add a description, additionally defining the time zone and the deviation of the computer's clock from the actual time, if applicable. If you also have a hash database of benign or malignant files, you can also specify the database. Then click on *Add Host*, and Autopsy will again display a confirmation page. Click on *OK* to confirm.

The *Host Gallery* is then displayed, allowing you to select a host for processing; again click *OK* to confirm before going on to add disk images. To do so, select *Add Image* and type the filename (see Figure 4).

This form is also used to specify whether Autopsy should add a symlink for the original file to the morgue directory, or if the image is to be copied or moved. You also need to specify the original mountpoint, the filesystem type and MD5 options. Autopsy calculates the MD5 checksum in every case; if you already know what this should be, you can type the value here, to allow Autopsy to verify it against the actual MD5 checksum.

Logbook November 7 2000

You can now go on to investigate the time-scale for filesystem modifications by selecting the *File Activity Time Lines* menu item. Doing so changes the appearance of the web application, splitting the window into two frames. The

Table 1: Challenge Partitions

Partition	Filesystem
/dev/hda8	/
/dev/hda1	/boot
/dev/hda6	/home
/dev/hda5	/usr
/dev/hda7	/var
/dev/hda9	swap

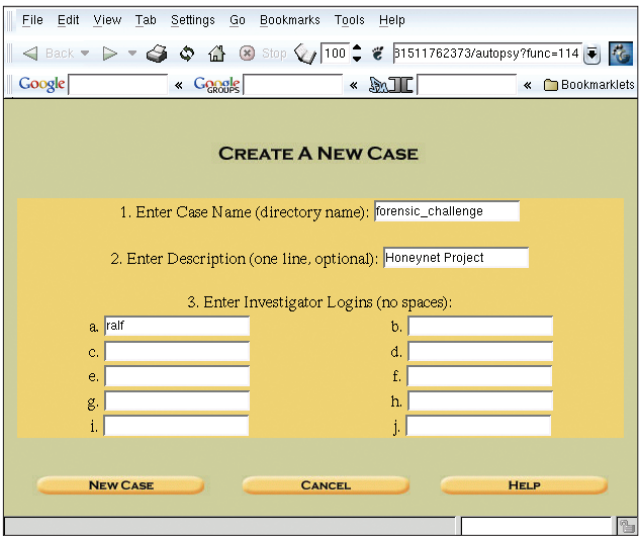

Figure 2: You need to specify a name for the new case (*forensic_challenge* in our example), and add logins for the investigator responsible for this case (*ralf*)

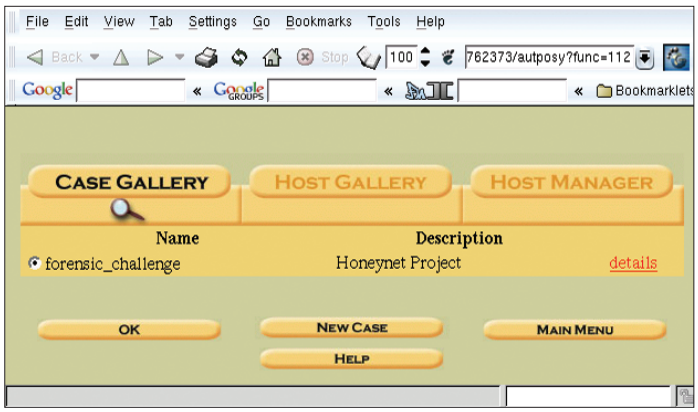

Figure 3: Autopsy organizes forensic investigations in cases. This allows you to switch to another case without having to restart the GUI

top frame shows typical steps as menu items, and the bottom frame is used for input and output.

Figure 5 shows how to create the so-called body file using the *Create Data File* menu item. This process can take a while, as Autopsy needs to call the Sleuthkit *fls* and *ils* commands. After completing these steps, Autopsy automatically computes an MD5 checksum which is used to perform integrity checks.

Autopsy then creates the timeline based on the body file. The *Create Timeline* menu item sensibly prompts you to restrict the time window to be investigated.

The description of the Forensic Challenges specifies November 7 2000 as the most likely date of the intrusion. For this example, the investigator will want to restrict the time window for more detailed investigations to the period between November 7 and November 9 2000. To allow Autopsy to replace the UID and GID with matching names when creating the timeline, you can stipulate the filesystem image containing the appropriate */etc/passwd* and */etc/group* files.

The timeline shows that the */etc/hosts.deny* file has been modified, reducing the filesize to zero bytes. A few minutes later, a tarball archive was installed in the */usr/man/.Ci/* directory (see Figure 6). This is the directory the forensic investigator will want to concentrate on later. A few seconds before this installation took place, inode 8 133 on *hda8* was deleted. The file size was 2 129 920 bytes, and the file was owned by a user called *drosen*. One file in the rootkit was deleted later, the file at inode 109 801 with a file size of 1 153 bytes.

The timeline also shows read access to a number of libraries. This would indicate that the intruder compiled one or more applications.

No Point Covering Your Tracks

After completing the installation the intruder deleted a large number of files. It would appear that these files were

Installing and Launching Autopsy

As Autopsy not only performs analysis, but also helps the investigator complete the inevitable paperwork that forensics entail, it organizes the task in hand as a collection of individual cases, assigning a directory to each case. It is a good idea to create a parent directory for the case directories before you launch into the installation process: */var/morgue* is a good name for your evidence locker.

Installing Autopsy is a little strange. After calling *make*, you will need to answer one or two questions. The installation script checks your computer for a current version of Sleuthkit before creating the configuration files.

When you launch the tool, by typing *./autopsy* in the source directory, the Autopsy Forensic Browser comes up showing its version number, a URL, and a message to the effect that you should allow this process to run during the analysis phase, and terminate the process by pressing [Ctrl]+[C] when finished.

You can now use any local Web server to support Autopsy access. Just type the URL that was shown previously in your browser's address box to do so. You can also use command line options to tell Autopsy to run on another port and IP address: *./autopsy Port-Number IP-Address*.

If you prefer to use the author's RPM packages [3] to install Sleuthkit and Autopsy, rather than the sources, you will note that the *autopsy* is in your default path. These packages use */var/morgue* as their evidence locker.

```
Terminal <3>
/opt/forensik/autopsy-1.73 # mkdir /var/morgue
/opt/forensik/autopsy-1.73 # make

   Autopsy Forensic Browser Installation

perl found: /usr/bin/perl
strings found: /usr/bin/strings
  Testing decimal offset flag of strings: PASS
  Testing non-object file arguments: PASS
grep found: /usr/bin/grep

Enter The Sleuth Kit Directory:
/opt/forensik/sleuthkit-1.64
  Sleuth Kit bin directory was found
  Required version found

Do you have the NIST National Software Reference Library (NSRL)? (y/n) [n]

Enter the Evidence Locker Directory (where cases will be saved):
/var/morgue
  /var/morgue already exists

Settings saved to conf.pl

/opt/forensik/autopsy-1.73 #
```

Figure 1: You can type *make* to compile and install Autopsy, but be prepared to answer a few questions. It makes sense to create a morgue directory before you start installing

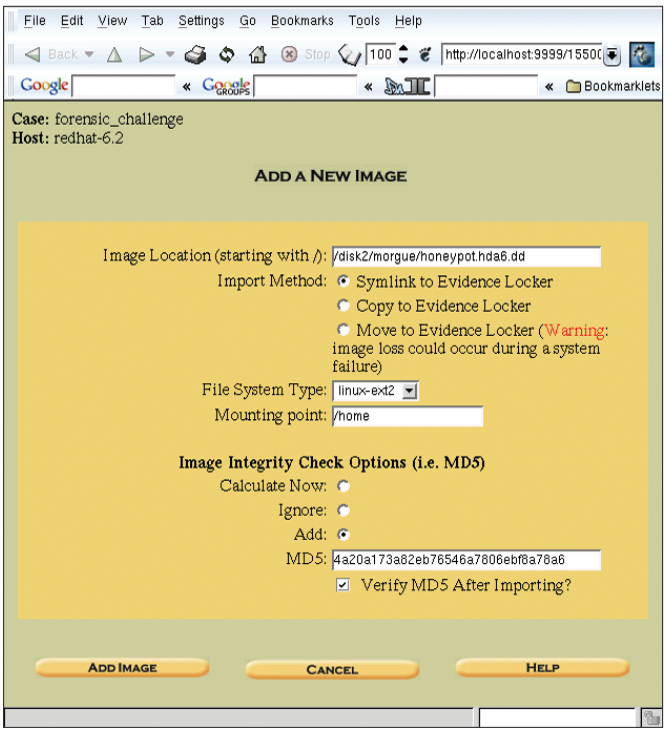

Figure 4: Before investigating a filesystem image, the investigator must first add the image to the case, specifying a checksum (MD5) to ensure that the image is not damaged

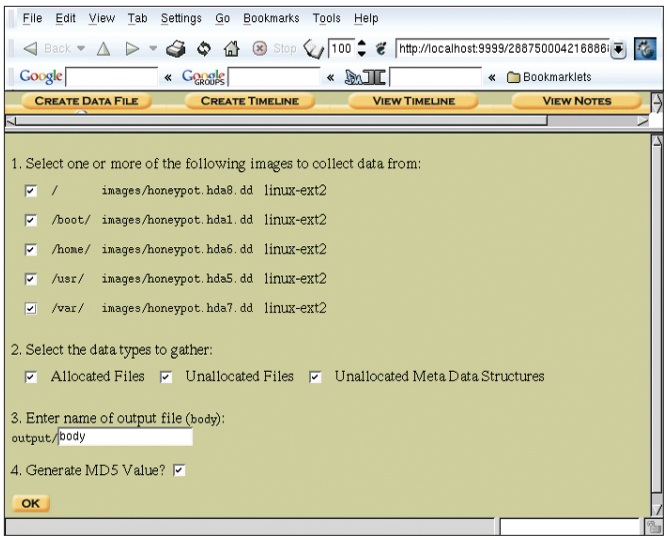

Figure 5: Autopsy creates a body file to store timelines for filesystem operations on the image. The body can apply to both deleted and existing files

created as part of the compilation process, and are no longer needed. After deleting the superfluous files, the intruder seems to have installed an SSH distribution (see Listing 1).

The timeline also indicates that the intruder used installation scripts for to gain access and install software. The timeline contains a number of entries concerning deleted files with names that support this assumption: *install-sshd1* and the like (see Listing 2).

The files shown here are not the only suspicious entries in the timeline. In the further course of the attack, the intruder seems to have planted an eggdrop and a copy of the Bitch X IRC client on the disk.

It is now the investigating admin's task to discover the nature of and motivation for the installed files. The installation scripts are typically a good place to start. The intruder has deleted these files, but Autopsy should have no trouble recovering them.

To recover the files, first close the timeline (*Close* top right) and select the */usr* partition. Then confirm by clicking *OK* to display a new view, where you then select *File Analysis*. This is the area where individual files, such as */usr/man/.Ci/install* named can be viewed (see Figure 7).

Hidden Processes

Fortunately, Autopsy also allows you to view other types of files. For example, */usr/man/.Ci/addps* contains a short script which is obviously used to hide processes normally displayed by the *top* or *ps* commands and thus to try and avoid detection. The attacker seems to have replaced the standard commands by variants from the rootkit. The content of the script is as follows:

```
#!/bin/sh
HIDE=$1
echo "hiding $HIDE from ps/top"
/bin/echo "2 $HIDE" >>/dev/ptyp
```

The modified *ps* and *top* commands need to read the */dev/ptyp* in order to hide these processes. The file contains the following entries:

```
2 slice2
2 snif
2 pscan
2 imp
```

Listing 1: SSH Installation

```
537       m.c  -/-rw------- root root    26570 /etc/ssh_host_key
880       .a.  -/-rw-r--r-- root root    26579 /etc/ssh_config
512       m.c  -/-rw------- root root     2048 /root/.ssh/random_seed
341       mac  -/-rw-r--r-- root root    26578 /etc/ssh_host_key.pub
...
604938 mac  -/-rws--x--x root root 109999 /usr/local/bin/ssh1
```

Listing 2: Installation scripts

```
1153   ..c -/-rwxr-xr-x 1010 users 109801 /usr/man/.Ci/install-sshd1 (deleted)
1076   ..c -/-rwxr-xr-x 1010 users 109802 /usr/man/.Ci/install-sshd (deleted)
80     .a. -/-rwxr-xr-x 1010 users 109803 /usr/man/.Ci/install-named (deleted)
71     ..c -/-rwxr-xr-x 1010 users 109867 /usr/man/.Ci/install-wu (deleted)
106    ..c -/-rwxr-xr-x 1010 users 109864 /usr/man/.Ci/install-statd (deleted)
...
```

```
3 qd
...
```

Analysis of the *ps* command with *strings* or the Autopsy front-end, as shown in Figure 8, indicates that this command contains the */dev/ptyp* string. This confirms the previous assumption since the original *ps* command does not read this file.

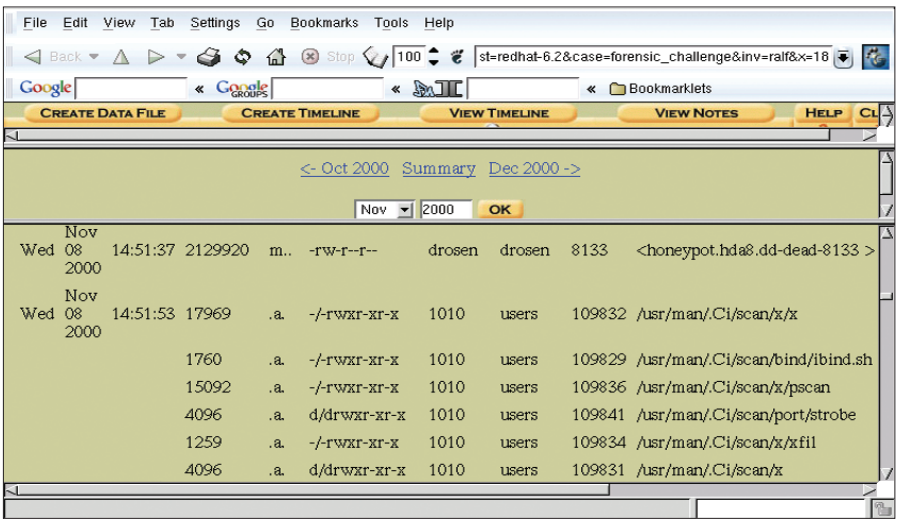

Figure 6: Autopsy reconstructing the installation of a rootkit. The columns contain the date and time, size, action (*a* for access, *m* for modify), rights, UID, GID, inode number, and name of the file in question

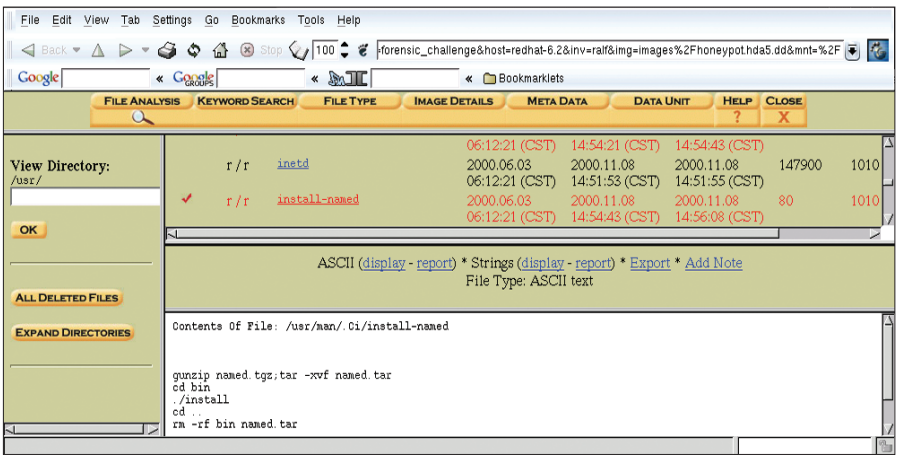

Figure 7: Autopsy's File Analysis module allows you to display any files on the filesystem, even deleted files (highlighted in red). The lower right panel shows the content of the deleted file *install-named*

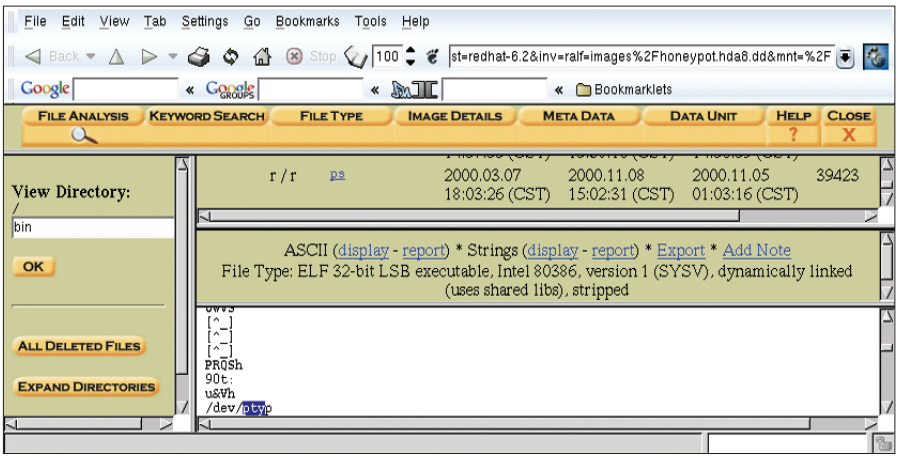

Figure 8: File analysis shows that the *ps* command contains a suspicious string, */dev/ptyp*. This file is not a device, but a list of processes the attacker wanted to hide

Trojans

The Secure Shell server installed by the attacker is another interesting file. */usr/local/sbin/sshd* contains a reference to */usr/tmp/nap*. The reference is easy to locate. Just look for the separator character */*. */usr/tmp* is a symbolic link to */var/tmp*. The */var/tmp/nap* file contains the following information:

```
username: root password: ➘
tw1Lightz0ne hostname: ➘
c871553-b.jffsn1.mo.home.com
```

In other words, the SSH server installed by the intruder stores any passwords it receives in cleartext format.

Enhanced Functions

Autopsy provides a number of additional functions, such as keyword searches, file sorting by type and direct access to file content. One major advantage of using Autopsy is the possibility to calculate an MD5 checksum on the fly, and add your own notes. An investigator would need to be extremely disciplined to achieve this using only the command line.

Autopsy's developers are currently working on index search routines for the keyword search feature. You only need to create the index file once, to speed up any ensuing searches. A search operation that takes 168 seconds at present, would take only 2 seconds using the new technique.

Conclusion

A combination of Sleuthkit with the Autopsy Forensic Browser provides an extremely powerful forensic analysis toolkit. Its features and facilities compare well with commercial tools. The fact that the program is Open Source allows investigators to trace the workings of the tool in detail. ■

INFO

[1] Ralf Spenneberg, "Sleuthkit, the Digital Forensic Toolkit", Linux Magazine Issue 35, October 2003, p54-59

[2] Autopsy Forensic Browser: *http://autopsy.sf.net*

[3] Autopsy and Sleuthkit RPM packages: *http://www.spenneberg.org/Forensics/*

[4] Forensic Challenge files: *http://project.honeynet.org/challenge/images.html*