Investigating Windows systems with Linux

WINDOW KIT

Max, Fotolia

A forensics expert explains how to extract interesting details from a confiscated Windows hard disk using standard Linux tools. BY HANS-PETER MERKEL AND MARKUS FEILNER

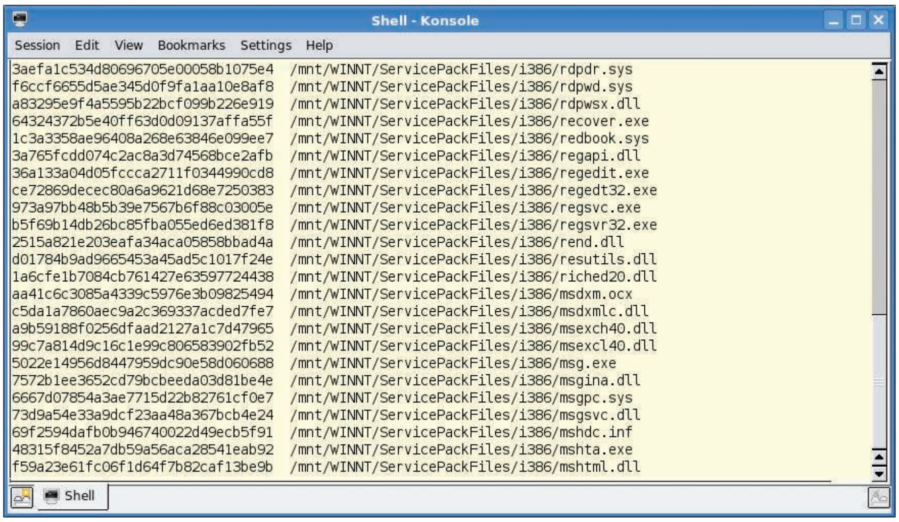
Criminals, intruders, and corporate saboteurs leave data behind on the hard disks of any computers they visit. Many of these computers are Windows systems, but you don't need Windows to extract valuable forensic information from a Windows hard disk. In this article, I will describe some simple techniques for getting forensic data from a Windows disk using Linux.

Before starting any forensic analysis, it is important to create a copy of the storage medium you will be investigating, either as a 1:1 copy or as an image or collection of images. You can copy the medium as a raw image (with *dd*) or use a format such as Expert Witness Format (EWF).

EWF is a proprietary format developed by Guidance Software [1] that is also supported by the X-Ways [2] commercial forensic tool. EWF images are compressed and thus are far smaller than raw images.

Linux tools such as Linux Encase (Linen) or Ewfacquire [3] will help you create an EWF image. Linen is included on the Helix forensics CD [4] as a free contribution by Guidance Software, but the *dd* tool, which is included on any Linux distribution, will normally do the trick. If you use *dd*, you can even launch a copy of the Windows system in a virtual environment such as VMware; EWF will not let you launch Windows without the proprietary add-on software because of its compressed format.

A command such as *dd if=/dev/sda of=win_hd.dd bs=4096 conv=noerror, sync* will create a *dd* image. Instead of



```
3aefa1c534d80696705e00058b1075e4  /mnt/WINNT/ServicePackFiles/i386/rdpdr.sys
f6ccf6655d5ae345d0f9fa1aa10e8af8  /mnt/WINNT/ServicePackFiles/i386/rdpwd.sys
a83295e9f4a5595b22bcf099b226e919  /mnt/WINNT/ServicePackFiles/i386/rdpwsx.dll
64324372b5e40ff63d0d09137affa55f  /mnt/WINNT/ServicePackFiles/i386/recover.exe
1c3a3358ae96408a268e63846e099ee7  /mnt/WINNT/ServicePackFiles/i386/redbook.sys
3a765fcdd074c2ac8a3d74568bce2afb  /mnt/WINNT/ServicePackFiles/i386/regapi.dll
36a133a04d05fccca2711f0344990cd8  /mnt/WINNT/ServicePackFiles/i386/regedit.exe
ce72869decec80a6a9621d68e7250383  /mnt/WINNT/ServicePackFiles/i386/regedt32.exe
973a97bb48b5b39e7567b6f88c03005e  /mnt/WINNT/ServicePackFiles/i386/regsvc.exe
b5f69b14db26bc85fba055ed6ed381f8  /mnt/WINNT/ServicePackFiles/i386/regsvr32.exe
2515a821e203eafa34aca05858bbad4a  /mnt/WINNT/ServicePackFiles/i386/rend.dll
d01784b9ad9665453a45ad5c1017f24e  /mnt/WINNT/ServicePackFiles/i386/resutils.dll
1a6cfe1b7084cb761427e63597724438  /mnt/WINNT/ServicePackFiles/i386/riched20.dll
aa41c6c3085a4339c5976e3b09825494  /mnt/WINNT/ServicePackFiles/i386/msdxm.ocx
c5da1a7860aec9a2c369337acded7fe7  /mnt/WINNT/ServicePackFiles/i386/msdxmlc.dll
a9b59188f0256dfaad2127a1c7d47965  /mnt/WINNT/ServicePackFiles/i386/msexch40.dll
99c7a814d9c16c1e99c806583902fb52  /mnt/WINNT/ServicePackFiles/i386/msexcl40.dll
5022e14956d8447959dc90e58d060688  /mnt/WINNT/ServicePackFiles/i386/msg.exe
7572b1ee3652cd79bcbeeda03d81be4e  /mnt/WINNT/ServicePackFiles/i386/msgina.dll
6667d07854a3ae7715d22b82761cf0e7  /mnt/WINNT/ServicePackFiles/i386/msgpc.sys
73d9a54e33a9dcf23aa48a367bcb4e24  /mnt/WINNT/ServicePackFiles/i386/msgsvc.dll
69f2594dafb0b946740022d49ecb5f91  /mnt/WINNT/ServicePackFiles/i386/mshdc.inf
48315f8452a7db59a56aca28541eab92  /mnt/WINNT/ServicePackFiles/i386/mshta.exe
f59a23e61fc06f1d64f7b82caf13be9b  /mnt/WINNT/ServicePackFiles/i386/mshtml.dll
```

Figure 1: Md5sum can create unique hashes for the files on a Windows filesystem. You can compare the values in the first column with the hash values for known files.

/dev/sda, just specify the disk you want to copy. The block size *bs = 4096* accelerates the copy. The *conv* parameter ensures that the copy will not terminate if it encounters defective sectors.

By entering *fdisk -lu*, you can obtain information on the disk image. The administrator simply needs to pass in the image name (Listing 1).

In Listing 1, the image contains a partition; the filesystem is NTFS. The *disktype* program from the standard Debian repositories provides more information (Listing 2). The partition obviously contains the Windows bootloader.

To access the filesystem, the administrator first needs to mount it. The partition starts with sector 63, which is normal for a hard disk. The exception to this is Microsoft's latest offspring, Windows Vista, in which the first sector is 2047. The *mount* command thus specifies the matching offset:

```
# losetup -o $((63*512)) ↩
/dev/loop0 U win_hd.dd
# mount -o ro,noatime,noexec ↩
/dev/loop0 /mnt
```

A quick glance at */mnt* reveals the startup files and filesystem of a Windows drive. A quick glance at the *boot.ini* file reveals that they belong to a Windows 2000 Server (Listing 3).

Seeking with Find

Criminal investigators often use the Linux *find* command with the *--exec* or *xargs* xx options to search for files with illegal content. After creating and mounting an image, *find /mnt -type f* will give you a detailed list of files. Because this approach does not take file names with blanks or non-standard characters into account, the investigator might opt for *find /mnt -type f -print0 | xargs -0 ls -al*.

Hash values help find identical and suspicious files on a system. You can create a hash automatically with a command such as *find /mnt -type f -print0 | xargs -0 md5sum*; you can even compare the hashes on the fly with existing reference material. However, it makes more sense to create a file containing hashes for all files (Figure 1).

Hashes Find Duplicates

In most cases, investigators already have hashes for files they want to find. Forensics experts compile databases with hashes of known files to help search for criminal material. If you just want to filter out Microsoft DLLs on a system you are investigating, this approach is useful.

A simple *grep* command will find any correlations between the subject of the investigation and your search targets.

The following command, from which we have removed the individual filenames, saves a list of hashes for existing files in a file called *big.txt*:

```
# find /mnt -type f -print0 | ↩
 xargs -0 md5sum | awk ↩
'{print $1}' |  sort -g | ↩
uniq > big.txt
```

Listing 1: Getting Information with fdisk

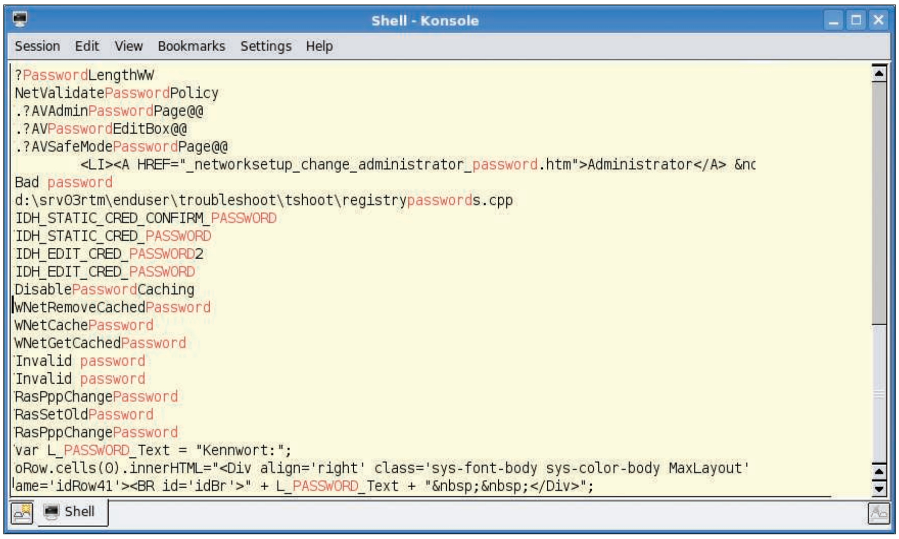
```
# fdisk -lu win_hd.dd
Disk win_hd.dd: 0 MB, 0 bytes
120 heads, 63 sectors/track, 0 cylinders, total 0 sectorsUnits =
  sectors of 1 * 512 = 512 bytes
Disk identifier: 0x840b840b
Device          Boot      Start         End    Blocks         Id
  System
win_hd.dd1      *         63          6327719163828+      7
  HPFS/NTFS
```

Listing 2: Disktype Data

```
# disktype win_hd.dd
--- win_hd.dd
Regular file, size 3.021 GiB (3243663360 bytes)
DOS/MBR partition map
Partition 1: 3.017 GiB  (3239760384 bytes, 6327657 sectors from 63,
  bootable)
  Type 0x07 (HPFS/NTFS)
  Windows NTLDR boot loader
  NTFS file system
    Volume size 3.017 GiB (3239759872 bytes, 6327656 sectors)
```

Listing 3: Windows Filesystem

```
# ls -l /mnt
Total 787024
-r-------- 1 root root    150528 2003-06-19 13:05 arcldr.exe
-r-------- 1 root root    163840 2003-06-19 13:05 arcsetup.exe
-r-------- 1 root root         0 2007-12-02 11:59 AUTOEXEC.BAT
-r-------- 1 root root       186 2007-12-02 11:43 boot.ini
-r-------- 1 root root         0 2007-12-02 11:59 CONFIG.SYS
dr-x------ 1 root root      4096 2007-12-02 14:14 Dokumente und
  Einstellungen
dr-x------ 1 root root     24576 2007-12-02 14:14 WINNT
(...)
# cat /mnt/boot.ini
 [boot loader]
 timeout=30
 default=multi(0)disk(0)rdisk(0)partition(1)\ WINNT
[operating systems]
 multi(0)disk(0)rdisk(0)partition(1)\WINNT="Microsoft Windows 2000
  Server" /fastdetect
(...)
```



```
?PasswordLengthWw
NetValidatePasswordPolicy
.?AVAdminPasswordPage@@
.?AVPasswordEditBox@@
.?AVSafeModePasswordPage@@
        <LI><A HREF="_networksetup_change_administrator_password.htm">Administrator</A> &nc
Bad password
d:\srv03rtm\enduser\troubleshoot\tshoot\registrypasswords.cpp
IDH_STATIC_CRED_CONFIRM_PASSWORD
IDH_STATIC_CRED_PASSWORD
IDH_EDIT_CRED_PASSWORD2
IDH_EDIT_CRED_PASSWORD
DisablePasswordCaching
WNetRemoveCachedPassword
WNetCachePassword
WNetGetCachedPassword
Invalid password
Invalid password
RasPppChangePassword
RasSetOldPassword
RasPppChangePassword
var L_PASSWORD_Text = "Kennwort:";
oRow.cells(0).innerHTML="<Div align='right' class='sys-font-body sys-color-body MaxLayout'
ame='idRow41'><BR id='idBr'>" + L_PASSWORD_Text + "&nbsp;&nbsp;</Div>";
```

Figure 2: Standard Linux tools in action. Cat, Strings, and Grep search the Windows partition for keywords and highlight any matches they find.

The *awk* command takes the hash values in the first column; *sort -g* sorts them, and *uniq* removes duplicate entries. An investigator who has stored the desired hash values in a file called *small.txt*, can then run *grep -f small.txt big.txt* to find duplicates, and thus any matching files.

Keyword Search

Forensic investigators search confiscated systems for specific keywords.

Creating a text file with the search keys for this purpose – such as *keywords.txt* – is a good idea. For example, you could add the words *password* and *secret*.

The command line

```
# cat win_hd.dd | strings | 
egrep -i --color -f keywords.txt
```

searches the complete Windows image for the keywords *password* and *secret* and highlights them in red in the output, as you can see in Figure 2. This approach is particularly interesting if you extend the search beyond the filesystem to other areas of a hard disk, such as:

- the swap file or hibernation file,
- unallocated areas of the disk,
- file slack data
- deleted files.

To find keywords stored in the 16-bit Unicode text that the Windows NT operating systems use, you must tell the *strings* command whether to perform a little – or big – endian [5] search. The required arguments are either *-eb* or *-el*. Listing 4 uses Ntfsundelete as an example of restoring files via inode allocations.

The Sleuth Kit

The Sleuth Kit [6] is one of the leading forensic tools. You'll find The Sleuth Kit packages in the standard Debian repository, and you can use Aptitude to install it. The Sleuth Kit mainly contains three tools that extend the functionality of *ls*; the names help explain the functions:

- *fls* lists files at file system level,
- *ils* lists files on the basis of inodes,
- *dls* restores deleted files.

In addition to these files are a couple of close relatives of the standard Unix tools *cat* (*icat*) and *find* (*ifind*), and statistics tools such as *istat*. The Sleuth Kit starts by creating a list of all files with time-

Listing 4: Restoring Files

```
# ntfsundelete -u -i11137 /dev/loop0
Inode    Flags  %age  Date           Size  Filename
-----------------------------------------------------
11137    FN..   0%    2003-06-19    50688  msiinst.exe
Undeleted 'msiinst.exe' successfully.
file msiinst.exe
msiinst.exe: MS-DOS executable PE  for MS Windows (DLL) (GUI) Intel
  80386 32-bit
```

Listing 5: File Timestamps

```
01 # fls -o 63 -m "C:" -r win_hd.dd > /tmp/body
02 # mactime -d -b /tmp/body
03 Thu Jun 19 2003 13:05:04,16656,m..,-/-rwxrwxrwx,0,0,315-128-3,C:/WINNT/system32/cdmodem.dll
05 Thu Jun 19 2003 13:05:04,11792,m..,-/-rwxrwxrwx,0,0,11267-128-3,C:/WINNT/ServicePackFiles/i386/partmgr.
  sys
07 Thu Jun 19 2003 13:05:04,7440,m..,-/-rwxrwxrwx,0,0,8093-128-3,C:/WINNT/ServicePackFiles/i386/
   bhp.dll
09 Thu Jun 19 2003 13:05:04,1011764,m..,-/-rwxrwxrwx,0,0,7102-128-3,C:/WINNT/system32/mfc42u.dll
11 Thu Jun 19 2003 13:05:04,65593,m..,-/-rwxrwxrwx,0,0,6552-128-3,C:/Programme/Outlook Express/
   csapi3t1.dll
13 Thu Jun 19 2003 13:05:04,122640,m..,-/-rwxrwxrwx,0,0,858-128-3,C:/WINNT/system32/idq.dll
14 Thu Jun 19 2003 13:05:04,166672,m..,-/-rwxrwxrwx,0,0,7178-128-3,C:/WINNT/system32/qcap.dll
15 Thu Jun 19 2003 13:05:04,65593,m..,-/-rwxrwxrwx,0,0,11555-128-3,C:/WINNT/Sersystem32/i386/csapi3t1.dll
```

stamp information (Listing 5). If you want a neatly grouped view with a timeline of the events, you can run *mactime -b /tmp/body*. To tell the tool to look for keywords in the deleted files on an NTFS partition:

```
# dls /dev/loop0 > unallocated

# cat unallocated | strings | ↵
egrep -i --color -f keywords.txt
```

The *dls* command converts the unallocated space into a file, which *cat* then pipes to *strings* and *egrep*.

File Slack

File slack [7] refers to data in the unused space on a filesystem. This effect occurs when you save, for example, a 2KB file on a filesystem with 4KB blocks. All popular Windows systems just pad the unused space with random data from RAM to fill up the blocks.

Tools such as dls from The Sleuth Kit, or bmap [8], let an investigator recreate data that the user sometimes didn't even knowingly store on their disk. Some investigators have used this approach to reconstruct incriminating emails.

dls with the *-s* option is particularly useful for this purpose:

```
# dls -s /dev/loop0 > fileslack
# cat fileslack | ↵
strings | egrep ↵
-i U--color -f keywords.txt
```

This gives the forensics expert the ability to search the file slack for keywords. According to a study [9], modern Linux filesystems are not affected by this problem; they pad the unused bytes with harmless zeros courtesy of */dev/zero*.

Restore Deleted Files

ntfsundelete, from the ntfsprogs package, gives any Linux admin the ability to restore deleted files on NTFS partitions. Before you run *ntfsundelete*, you first need to release the */dev/loop0* device, typically by issuing a *umount /mnt*.

THE AUTHOR

Hans-Peter Merkel has been an active member of the open source forensics community for many years. He trains criminal investigators in Germany and Tanzania, and he is one of the founders of Freioss and Linux4afrika.

Without specifying any additional options, *ntfsundelete /dev/loop0* just outputs a list of all undeletable files (Figure 3). The example in Figure 3 undeletes the *msiinst.exe* file on inode 11137.

Files existing on the hard disk could provide much user information. Both Microsoft's Internet Explorer and Firefox store their history on the filesystem. The investigator needs to install two programs to analyze the information:

- Pasco [10] for Internet Explorer
- Mork.pl [11] for Mozilla Firefox

Listing 6 shows a typical analysis se-

Listing 6: Browser History

```
# mount -o ro,noatime,noexec /dev/loop0 /mnt
# find /mnt -iname "index.dat" -exec pasco '{}' ';'
TYPE    URL     MODIFIED TIME   ACCESS TIME     FILENAME       DIRECTORY      HTTP  HEADERS
URL     http://www.google.de/favicon.ico       06/07/2006 21:35:34     12/02/2007
12:14:28     favicon[1].ico  NGORCTFI       HTTP/1.1 200 OK  Content-Type: image/x-icon
Content-Length: 1406    ~U:administrator
REDR    http://msn.ivwbox.de/cgi-bin/ivw/CP/MSN01000000;?r=
        12/02/2007 12:11:32     12/02/2007 12:11:32
URL     Visited: Administrator@http://www.google.de     12/02/2007 12:14:28
URL     Visited: Administrator@http://www.msn.de       12/02/2007 14:33:54     12/02/2007
14:33:54
# find /mnt -iname "history.dat" -exec mork.pl '{}' ';'
1202727704      1       http://www.linux4afrika.de/index.php?id=155&L=1
1202727670      1       http://www.linux4afrika.de/index.php?id=154&L=1
1202727641      1       http://www.linux4afrika.de/index.php?id=60&L=1
1202727641      2       http://www.linux4afrika.de/
1202727555      1       http://n-tv.de/916916.html
1202726960      1       http://n-tv.de/916917.html
1202726892      1       http://n-tv.de/916908.html
1202726827      3       http://n-tv.de/
1202726394      2       http://www.linux-magazine.com/
1202726204      2       http://www.google.de/
# find /mnt -iname "history.dat" -exec mork.pl '{}' ';' | awk '{print strftime("%F,%R",$1),$2,$3}'
2008-02-11 11:40 1 http://www.linux-magazin.com/heft_abo/ausgaben/2008/03/zwerg_am_druecker
2008-02-11 11:39 2 http://www.linux-magazine.com/
2008-02-11 11:36 2 http://www.google.com/
(...)
```

quence: Internet Explorer stores information for each profile in files titled *index.dat*. Running a *find* command against the file gives the investigator a list of the pages accessed in the browser.

Revealing Mozilla

Firefox stores its data in *history.dat*. The first column contains the date and time information in Unix timestamp format. The third command in Listing 6 converts this to a human-compatible format. Dumphive [12] provides an approach to making the registry on a Windows system more easily readable (Figure 4). The *dumphive /mnt/WINNT/system32/config/system system.txt* command stores the register in a separate text file, which the investigator can probe using Unix text tools.

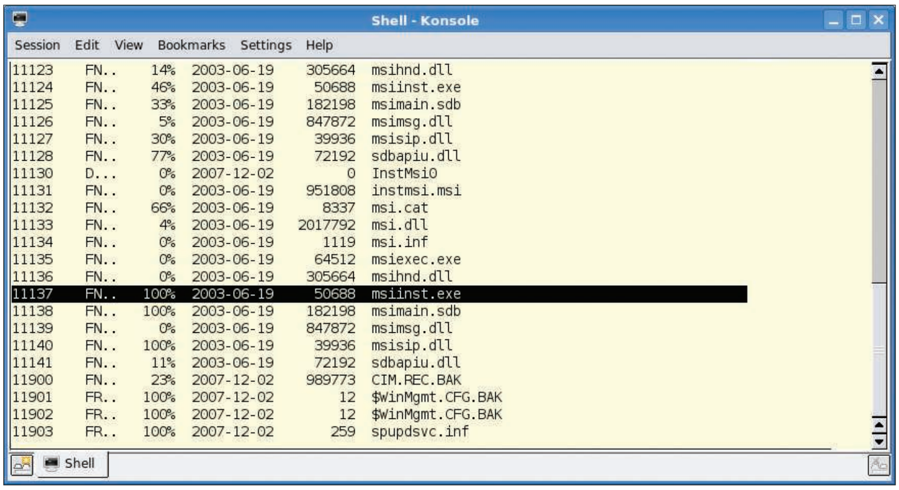
Windows Passwords

Access to the system is interesting in its own right, but discovering the user's passwords often opens up other vectors to the investigators, as most users don't bother changing their passwords when they log in to various websites and services. On top of this, passwords give the forensics investigator the ability to log in to a virtualized image system on VMware and to investigate various system logs and files.

In addition to employing brute force attacks and tools like John the Ripper, which use dictionaries or rainbow tables, the Linux admin can turn to other tools such as Bkhive, Samdump2, and Ophcrack [13].

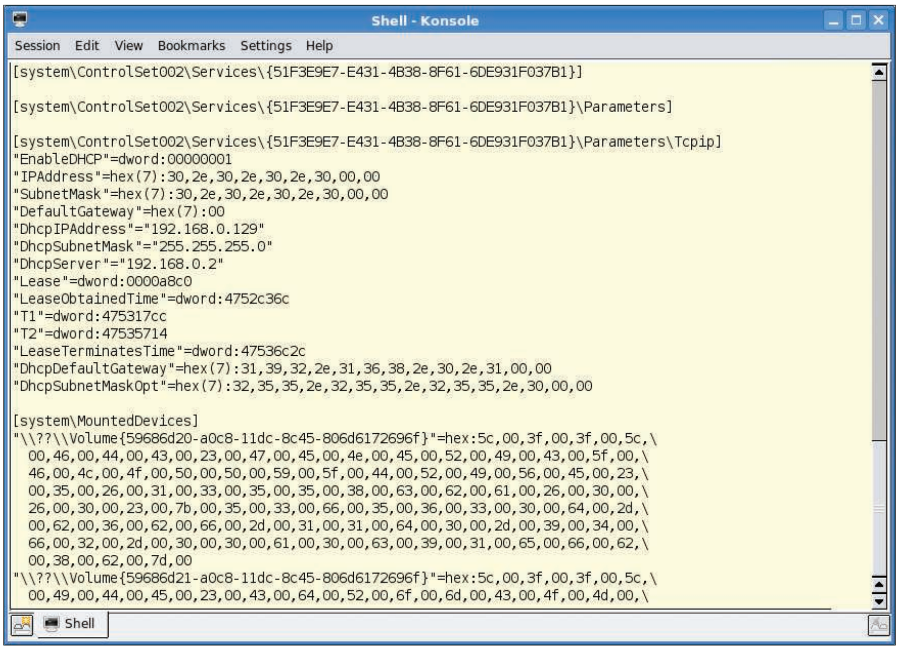
Extracting local passwords from a SAM file is not hard, as used by Windows NT-based operating systems, especially if you combine various tools. For example, John the Ripper automatically detects a Windows SAM file if you feed it to the tool. The Microsoft-specific password genus is useful here: Although Windows credentials can be up to 14 characters, the system splits them into two strings of seven characters each. This is a useful contribution by the manufacturer that makes it easier for investigators to break passwords without extreme number crunching.

As of Windows Vista, Microsoft closed this gap and replaced the Lanmanager hashes with NT hashes. XP admins can set this up manually; investigators on Linux have to run *dumphive* to check whether the Registry contains an entry



```
11123    FN..    14%  2003-06-19    305664  msihnd.dll
11124    FN..    46%  2003-06-19     50688  msiinst.exe
11125    FN..    33%  2003-06-19    182198  msimain.sdb
11126    FN..     5%  2003-06-19    847872  msimsg.dll
11127    FN..    30%  2003-06-19     39936  msisip.dll
11128    FN..    77%  2003-06-19     72192  sdbapiu.dll
11130    D...     0%  2007-12-02         0  InstMsi0
11131    FN..     0%  2003-06-19    951808  instmsi.msi
11132    FN..    66%  2003-06-19      8337  msi.cat
11133    FN..     4%  2003-06-19   2017792  msi.dll
11134    FN..     0%  2003-06-19      1119  msi.inf
11135    FN..     0%  2003-06-19     64512  msiexec.exe
11136    FN..     0%  2003-06-19    305664  msihnd.dll
11137    FN..   100%  2003-06-19     50688  msiinst.exe
11138    FN..   100%  2003-06-19    182198  msimain.sdb
11139    FN..     0%  2003-06-19    847872  msimsg.dll
11140    FN..   100%  2003-06-19     39936  msisip.dll
11141    FN..    11%  2003-06-19     72192  sdbapiu.dll
11900    FN..    23%  2007-12-02    989773  CIM.REC.BAK
11901    FR..   100%  2007-12-02        12  $WinMgmt.CFG.BAK
11902    FR..   100%  2007-12-02        12  $WinMgmt.CFG.BAK
11903    FR..   100%  2007-12-02       259  spupdsvc.inf
```

Figure 3: Ntfsundelete shows deleted files that can be undeleted. The first column contains the inode number required to restore the file on the filesystem in the image file.



```
[system\ControlSet002\Services\{51F3E9E7-E431-4B38-8F61-6DE931F037B1}]

[system\ControlSet002\Services\{51F3E9E7-E431-4B38-8F61-6DE931F037B1}\Parameters]

[system\ControlSet002\Services\{51F3E9E7-E431-4B38-8F61-6DE931F037B1}\Parameters\Tcpip]
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,2e,30,2e,30,2e,30,00,00
"SubnetMask"=hex(7):30,2e,30,2e,30,2e,30,00,00
"DefaultGateway"=hex(7):00
"DhcpIPAddress"="192.168.0.129"
"DhcpSubnetMask"="255.255.255.0"
"DhcpServer"="192.168.0.2"
"Lease"=dword:0000a8c0
"LeaseObtainedTime"=dword:4752c36c
"T1"=dword:475317cc
"T2"=dword:47535714
"LeaseTerminatesTime"=dword:47536c2c
"DhcpDefaultGateway"=hex(7):31,39,32,2e,31,36,38,2e,30,2e,31,00,00
"DhcpSubnetMaskOpt"=hex(7):32,35,35,2e,32,35,35,2e,32,35,35,2e,30,00,00

[system\MountedDevices]
"\\??\\Volume{59686d20-a0c8-11dc-8c45-806d6172696f}"=hex:5c,00,3f,00,3f,00,5c,\
  00,46,00,44,00,43,00,23,00,47,00,45,00,4e,00,45,00,52,00,49,00,43,00,5f,00,\
  46,00,4c,00,4f,00,50,00,50,00,59,00,5f,00,44,00,52,00,49,00,56,00,45,00,23,\
  00,35,00,26,00,31,00,33,00,35,00,35,00,38,00,63,00,62,00,61,00,26,00,30,00,\
  26,00,30,00,23,00,7b,00,35,00,33,00,66,00,35,00,36,00,33,00,30,00,64,00,2d,\
  00,62,00,36,00,62,00,66,00,2d,00,31,00,31,00,64,00,30,00,2d,00,39,00,34,00,\
  66,00,32,00,2d,00,30,00,30,00,61,00,30,00,63,00,39,00,31,00,65,00,66,00,62,\
  00,38,00,62,00,7d,00
"\\??\\Volume{59686d21-a0c8-11dc-8c45-806d6172696f}"=hex:5c,00,3f,00,3f,00,5c,\
  00,49,00,44,00,45,00,23,00,43,00,64,00,52,00,6f,00,6d,00,43,00,4f,00,4d,00,\
```

Figure 4: Dumphive converts the Windows Registry to cleartext. Search tools such as grep can then help you reconstruct critical settings, for example, the password hash or IP data.

INFO

[1] Guidance Software: *http://www.guidancesoftware.com*

[2] X-Ways: *http://www.x-ways.net/corporate/index-m.html*

[3] Ewfacquire: *https://www.uitwisselplatform.nl/projects/libewf*

[4] Helix: *http://www.e-fense.com/helix*

[5] Endianness: *http://en.wikipedia.org/wiki/Endianness*

[6] The Sleuth Kit: *http://sleuthkit.org*

[7] Wikipedia on file slack: *http://en.wikipedia.org/wiki/File-Slack*

[8] bmap: *http://www.packetstormsecurity.org/linux/security/bmap-1.0.17.tar.gz*

[9] File slack analysis on Linux: *http://www.woerter.at/dud/stuff/fileslack.pdf*

[10] Pasco download: *http://downloads.sourceforge.net/odessa/pasco_20040505_1.tar.gz?modtime=1083715200&big_mirror=0*

[11] Mork.pl: *http://www.jwz.org/hacks/mork.pl*

[12] Dumphive: *http://v4.guadalinex.org/guadalinex-toro/pool/main/d/dumphive/dumphive_0.0.3-1_i386.deb*

[13] Ophcrack and Ophcrack Live CD: *http://ophcrack.sourceforge.net*

[14] Foundstone Forensic Tools: *http://www.foundstone.com/us/resources-free-tools.asp*

that sets *HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Control/Lsa* to *1*. If so, your only option might be to open your wallet and buy an 8.5GB dual-layer DVD with commercial LM or NT Rainbow Tables.

Like the other tools referred to in this article, Ophcrack is included in the Debian repositories. The Ophcrack utility requires rainbow tables and the hashes from the Windows machine. After completing the install, the investigator can work conveniently with the GUI and just double-click to decipher a user's password (Figure 5).

Domain Controllers

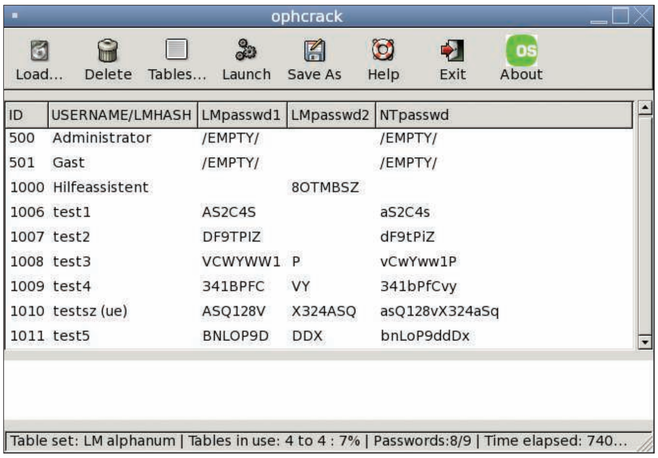
A different approach is required for Windows systems that log in to domain controllers; in this case, the credentials are not stored locally on the client. However, in many cases it is sufficient to run a network sniffer to capture the login exchange, identify the relevant data packets, save them to a file, and then feed a dump to Ophcrack. Of course, this is more complex, and you do need live access to the network.

Comparatively Simple

With the addition of a couple of extra packages, the Windows world is wide open to an investigator running Linux. If you need more of this good thing, take a look at the free forensic tools by Foundstone [14]. These tools give investigators the ability to restore cookies, long-gone entries from the Windows trash can, and many other things.

Experienced Linux users might find the shell approach refreshing, but some users will prefer to avoid the complex command-line syntax. The learning curve for Linux newcomers will likely be steeper for open source tools compared with more expensive commercial products. The winner in the usability stakes has to be the fully automated Ophcrack Live CD, which removes the need for users to type pesky shell commands and displays the local user's Windows passwords shortly after booting.

When we tested this on an XP system (SP2), the CD took just 280 seconds to discover the credentials of the five user accounts (which included up to 14 characters; see Figure 5). The live Linux version on the CD includes just the tables for alphanumeric passwords without non-standard characters. If you want more, you will have to invest in the commercial Rainbow Tables. ■



ID	USERNAME/LMHASH	LMpasswd1	LMpasswd2	NTpasswd
500	Administrator	/EMPTY/		/EMPTY/
501	Gast	/EMPTY/		/EMPTY/
1000	Hilfeassistent		8OTMBSZ	
1006	test1	AS2C4S		aS2C4s
1007	test2	DF9TPIZ		dF9tPiZ
1008	test3	VCWYWW1	P	vCwYww1P
1009	test4	341BPFC	VY	341bPfCvy
1010	testsz (ue)	ASQ128V	X324ASQ	asQ128vX324aSq
1011	test5	BNLOP9D	DDX	bnLoP9ddDx

Table set: LM alphanum | Tables in use: 4 to 4 : 7% | Passwords:8/9 | Time elapsed: 740...

Figure 5: Double-click to crack a Windows user's password. An automated Live CD is also available.