

# Introducción a la Informática Forense



Francisco Lázaro Domínguez



Ra-Ma®

# **Introducción a la Informática Forense**

# Introducción a la Informática Forense

*Francisco Lázaro Domínguez*



La ley prohíbe  
copiar o imprimir este libro

INTRODUCCIÓN A LA INFORMÁTICA FORENSE  
© Francisco Lázaro Domínguez

© De la Edición Original en papel publicada por Editorial RA-MA  
ISBN de Edición en Papel: 978-84-9964-209-3

Todos los derechos reservados © RA-MA, S.A. Editorial y Publicaciones, Madrid, España.

MARCAS COMERCIALES. Las designaciones utilizadas por las empresas para distinguir sus productos (hardware, software, sistemas operativos, etc.) suelen ser marcas registradas. RA-MA ha intentado a lo largo de este libro distinguir las marcas comerciales de los términos descriptivos, siguiendo el estilo que utiliza el fabricante, sin intención de infringir la marca y solo en beneficio del propietario de la misma. Los datos de los ejemplos y pantallas son ficticios a no ser que se especifique lo contrario.

RA-MA es una marca comercial registrada.

Se ha puesto el máximo empeño en ofrecer al lector una información completa y precisa. Sin embargo, RA-MA Editorial no asume ninguna responsabilidad derivada de su uso ni tampoco de cualquier violación de patentes ni otros derechos de terceras partes que pudieran ocurrir. Esta publicación tiene por objeto proporcionar unos conocimientos precisos y acreditados sobre el tema tratado. Su venta no supone para el editor ninguna forma de asistencia legal, administrativa o de ningún otro tipo. En caso de precisarse asesoría legal u otra forma de ayuda experta, deben buscarse los servicios de un profesional competente.

Reservados todos los derechos de publicación en cualquier idioma.

Según lo dispuesto en el Código Penal vigente ninguna parte de este libro puede ser reproducida, grabada en sistema de almacenamiento o transmitida en forma alguna ni por cualquier procedimiento, ya sea electrónico, mecánico, reprográfico, magnético o cualquier otro sin autorización previa y por escrito de RA-MA; su contenido está protegido por la Ley vigente que establece penas de prisión y/o multas a quienes, intencionadamente, reprodujeren o plagiaren, en todo o en parte, una obra literaria, artística o científica.

Editado por:  
RA-MA, S.A. Editorial y Publicaciones  
Calle Jarama, 33, Polígono Industrial IGARSA  
28860 PARACUELLOS DE JARAMA, Madrid  
Teléfono: 91 658 42 80  
Fax: 91 662 81 39  
Correo electrónico: [editorial@ra-ma.com](mailto:editorial@ra-ma.com)  
Internet: [www.ra-ma.es](http://www.ra-ma.es) y [www.ra-ma.com](http://www.ra-ma.com)

Maquetación: Gustavo San Román Borrueto  
Diseño Portada: Antonio García Tomé

ISBN: 978-84-9964-426-4

E-Book desarrollado en España en septiembre de 2014.

*Este libro está dedicado al Informático Desconocido, ese oscuro personaje que lo aprendió todo por su cuenta, montó una tienda de ordenadores, cobró a sus clientes las horas trabajadas para solucionar fallos propios y quebró en la crisis de las puntocom del año 2000. Que tras haberse sobrepuerto al desánimo causado por todos estos fracasos desarrolló páginas web, puso en marcha un sitio de comercio electrónico, escribió en blogs y redes sociales y formó parte de una plataforma de software libre. Y todo esto lo hizo alternando mal pagados trabajos de comercial, agente de seguros o incluso vigilante jurado, y en medio de unos y otros una separación matrimonial y el embargo de su coche por impago de multas de aparcamiento. Es cierto que Alan Turing, Bill Gates y Linus Torvalds plantaron la bandera en lo alto del montículo. Pero el despegue de las Tecnologías de la Información difícilmente habría sido posible sin este individuo anónimo y mal vestido, que no sabe hacerse el nudo de la corbata, y que en el ordenador de sobremesa halló no solo un modesto horizonte complementario a su currículum profesional y una fuente esporádica de ingresos, sino a veces también una razón de existir. Hora va siendo de que alguien se acuerde de este heroico zoquete. ¿No se hacen estatuas a soldados de infantería sin suerte y sin rostro que sufrieron el rigor de la vida en las trincheras y (en el mejor de los casos) volvieron a casa con las manos vacías, solo para que un mariscal pudiera hallar su lugar en los libros de historia y retirarse con una buena pensión? Pues entonces. A falta de bronce para fundir una placa y un pedestal hecho con residuo de obra para engastarla, aquí van estas líneas.*

# ÍNDICE

---

<b>INTRODUCCIÓN .....</b>	<b>17</b>
<b>CAPÍTULO 1. SE HA ESCRITO UN CRIMEN DIGITAL.....</b>	<b>23</b>
1.1 UN DÍA COMO OTRO CUALQUIERA .....	23
1.2 INTERVENCIÓN.....	25
1.2.1 Los primeros en llegar.....	25
1.2.2 Apagado sin más .....	25
1.2.3 Objetos intervenidos.....	26
1.3 LA AMENAZA DIGITAL .....	27
1.3.1 El delito informático.....	28
1.3.2 Evaluación del riesgo .....	29
1.3.3 Los motivos del agresor .....	30
1.3.4 Amenazas internas y externas .....	31
1.4 DINÁMICA DE UNA INTRUSIÓN .....	32
1.4.1 Footprinting.....	33
1.4.2 Escaneo de puertos y protocolos .....	34
1.4.3 Enumeración.....	35
1.4.4 Penetración y despliegue de exploits .....	35
1.4.5 Puertas traseras .....	36
1.4.6 Borrando huellas .....	36
<b>CAPÍTULO 2. LA INVESTIGACIÓN FORENSE .....</b>	<b>37</b>
2.1 ETAPAS DE UNA INVESTIGACIÓN FORENSE .....	37
2.1.1 Adquisición (Imaging) .....	38

2.1.2 Análisis.....	39
2.1.3 Presentación .....	39
2.1.4 La línea de tiempo.....	40
2.2 REQUISITOS DE LA INVESTIGACIÓN FORENSE .....	41
2.2.1 Aceptabilidad .....	41
2.2.2 Integridad .....	41
2.2.3 Credibilidad.....	42
2.2.4 Relación causa-efecto.....	42
2.2.5 Carácter repetible .....	42
2.2.6 Documentación.....	42
2.3 VALORACIÓN JURÍDICA DE LA PRUEBA DIGITAL.....	44
2.3.1 Interés legal de la prueba.....	44
2.3.2 Prueba física y prueba personal.....	44
2.3.3 Cualificación del investigador forense.....	45
2.3.4 La adquisición: fase crucial.....	45
<b>CAPÍTULO 3. SOPORTES DE DATOS.....</b>	<b>47</b>
3.1 PROCEDIMIENTOS DE ADQUISICIÓN .....	47
3.1.1 EnCase & Linen .....	49
3.1.2 dd.....	50
3.1.3 dcfldd, dc3dd y ddrescue.....	52
3.1.4 AIR .....	53
3.1.5 Adquisición por hardware .....	54
3.1.6 MD5 y SHA .....	55
3.1.7 Cálculo de MD5 con Linux .....	56
3.2 DISCOS, PARTICIONES Y SISTEMAS DE ARCHIVOS .....	56
3.2.1 NTFS .....	57
3.2.2 FAT .....	58
3.2.3 ext2, ext3, ext4 .....	60
3.2.4 HFS, HFS+, JFS, ReiserFS, etc.....	61
3.3 MODELO DE CAPAS .....	62
3.3.1 Nivel 1: dispositivos físicos .....	62
3.3.2 Nivel 2: volúmenes y particiones .....	63
3.3.3 Nivel 3: sistemas de archivos .....	64
3.3.4 Nivel 4: bloques de datos .....	64
3.3.5 Nivel 5: metadatos .....	65
3.3.6 Nivel 6: nombre de archivo .....	66
3.3.7 Nivel 7: journaling .....	66

3.4 RECUPERACIÓN DE ARCHIVOS BORRADOS.....	67
3.4.1 Dinámica del borrado de archivos.....	68
3.4.2 Sector/, cluster/ y file slack.....	69
3.5 ANÁLISIS DE UNA IMAGEN FORENSE CON TSK.....	70
3.5.1 Componentes de TSK.....	71
3.5.2 Adquisición de un soporte de datos .....	73
3.5.3 Instalación de TSK.....	74
3.5.4 Análisis de la imagen .....	74
3.5.5 Análisis del sistema de archivos .....	76
3.5.6 Listado de archivos.....	79
3.5.7 Recuperando archivos borrados .....	81
3.6 ANÁLISIS DE ARCHIVOS.....	82
3.6.1 Firmas características .....	82
3.6.2 Documentos.....	84
3.6.3 Archivos gráficos .....	85
3.6.4 Multimedia .....	87
3.6.5 Archivos ejecutables .....	89
3.6.6 Exclusión de archivos conocidos .....	90
3.7 DATA CARVING .....	91
3.7.1 Cuando todo lo demás falla.....	92
3.7.2 Extracción de archivos .....	92
<b>CAPÍTULO 4. ANÁLISIS FORENSE DE SISTEMAS MICROSOFT</b>	
<b>WINDOWS.....</b>	<b>95</b>
4.1 RECOPILANDO INFORMACIÓN VOLÁTIL.....	96
4.1.1 Fecha y hora del sistema .....	96
4.1.2 Conexiones de red abiertas.....	97
4.1.3 Puertos TCP y UDP abiertos.....	98
4.1.4 Ejecutables conectados a puertos TCP y UDP.....	99
4.1.5 Usuarios conectados al sistema .....	100
4.1.6 Tabla de enrutamiento interna.....	101
4.1.7 Procesos en ejecución.....	102
4.1.8 Archivos abiertos.....	103
4.2 ANÁLISIS FORENSE DE LA RAM .....	103
4.2.1 Captura de RAM completa con dd.....	104
4.2.2 Volcado de RAM .....	105
4.3 ADQUISICIÓN DE SOPORTES .....	106
4.3.1 Adquisición con EnCase .....	107

4.3.2 Adquisición con FTK Imager.....	108
4.3.3 Otros métodos .....	109
4.4 ANÁLISIS POST MORTEM .....	110
4.4.1 Análisis con EnCase.....	110
4.4.2 AccessData FTK .....	112
4.4.3 Captain Nemo.....	115
4.4.4 Mount Image Pro .....	116
4.4.5 FileDisk .....	116
4.5 INVESTIGACIÓN DEL HISTORIAL DE INTERNET .....	117
4.5.1 Microsoft Internet Explorer.....	118
4.5.2 X-Ways Trace .....	119
4.5.3 iehist .....	120
4.5.4 Historial de navegación en Mozilla/Firefox.....	120
4.5.5 Chrome .....	122
4.6 LA PAPELERA DE RECICLAJE .....	122
4.6.1 Análisis de la papelera con Rifiuti .....	124
4.6.2 Funcionamiento de la papelera en Windows Vista/7.....	124
4.7 COOKIES .....	125
4.8 CORREO ELECTRÓNICO .....	126
4.8.1 Formatos PST y DBX Folders .....	127
4.8.2 Otros clientes de correo .....	128
4.8.3 Paraben's E-Mail Examiner .....	128
4.9 BÚSQUEDA DE CARACTERES .....	129
4.9.1 SectorSpy, Disk Investigator y Evidor .....	129
4.9.2 X-Ways Forensics .....	130
4.10 METADATOS .....	131
4.10.1 Cómo visualizar los metadatos de un documento .....	132
4.10.2 Metadata Assistant .....	133
4.10.3 FOCA .....	134
4.10.4 Metadatos EXIF .....	135
4.11 ANÁLISIS DE PARTICIONES NTFS Y FAT .....	136
4.11.1 Runtime DiskExplorer .....	137
4.11.2 Recuperación de archivos borrados .....	138
4.11.3 Runtime GetDataBack .....	138
4.11.4 EasyRecovery Professional .....	139
4.11.5 R-Studio .....	140

4.12 EL REGISTRO DE WINDOWS .....	141
4.12.1 Estructura y archivos del Registro .....	142
4.12.2 Análisis off line con Windows Registry Recovery .....	143
4.12.3 RegRipper.....	144
<b>CAPÍTULO 5. ANÁLISIS FORENSE DE SISTEMAS LINUX/UNIX.....</b>	<b>145</b>
5.1 HERRAMIENTAS DE CÓDIGO LIBRE .....	146
5.1.1 ¿Qué es exactamente el código libre?.....	146
5.1.2 Linux en la investigación forense .....	146
5.1.3 Poniendo en marcha una estación de trabajo con Linux .....	147
5.1.4 Descarga, compilación e instalación de herramientas.....	148
5.1.5 Montaje automático de particiones .....	148
5.2 ESTRUCTURA TÍPICA DE UN SISTEMA LINUX.....	151
5.2.1 Arquitectura y sistemas de archivos.....	151
5.2.2 Jerarquía de directorios .....	152
5.2.3 Archivos y permisos.....	154
5.2.4 Marcas de tiempo .....	157
5.3 INFORMACIÓN VOLÁTIL .....	157
5.3.1 Fecha y hora del sistema .....	158
5.3.2 Información de interés.....	158
5.3.3 Puertos y conexiones abiertas .....	160
5.3.4 Procesos en ejecución.....	161
5.4 ADQUISICIÓN FORENSE.....	161
5.4.1 Adquisición con dd.....	162
5.4.2 Adepto .....	163
5.5 ANÁLISIS .....	165
5.5.1 La línea de tiempo .....	165
5.5.2 Herramientas para elaborar una línea de tiempo.....	166
5.5.3 Recuperación de archivos borrados .....	167
5.6 OTRAS HERRAMIENTAS .....	168
5.6.1 Chkrootkit y Rkhunter.....	168
5.6.2 Md5deep.....	170
<b>CAPÍTULO 6. REDES E INTERNET.....</b>	<b>173</b>
6.1 COMPONENTES DE UNA RED .....	173
6.1.1 Visión general de una red corporativa .....	174
6.1.2 Archivos de registro .....	174
6.1.3 Preservación de elementos de evidencia en redes.....	175

6.1.4 Siguiendo pistas.....	177
6.2 PROTOCOLOS .....	178
6.2.1 Capa de transporte: TCP .....	180
6.2.2 Puertos .....	181
6.2.3 Capa de red: IP .....	182
6.2.4 Enrutamiento .....	184
6.2.5 Capa de enlace de datos: interfaces Ethernet.....	185
6.2.6 Protocolos de nivel superior: HTTP y SMB .....	187
6.3 ANALIZANDO EL TRÁFICO DE RED .....	190
6.3.1 Wireshark .....	190
6.3.2 Captura de tráfico: hubs, mirroring, bridges .....	191
6.3.3 Utilización de Wireshark.....	193
6.3.4 Un ejemplo práctico .....	196
6.4 COMPROBACIÓN DE DIRECCIONES IP .....	199
6.4.1 Herramientas de traza de red.....	199
6.4.2 Whois o quién es quién en Internet .....	200
6.4.3 Ping/fping .....	202
6.4.4 Traceroute/tracert .....	203
6.5 CORREO ELECTRÓNICO.....	204
6.5.1 Cabeceras e-mail .....	205
6.5.2 Estructura típica de un encabezado .....	205
<b>CAPÍTULO 7. INVESTIGACIÓN FORENSE DE DISPOSITIVOS MÓVILES.....</b>	<b>209</b>
7.1 TELÉFONOS MÓVILES INTELIGENTES .....	210
7.1.1 Smartphones: pasaporte al siglo XXI.....	210
7.1.2 Hardware .....	212
7.1.3 Software .....	213
7.1.4 Información obtenible .....	215
7.2 INVESTIGACIÓN FORENSE DEL APPLE IPHONE.....	219
7.2.1 Consideraciones generales .....	219
7.2.2 Adquisición del iPhone mediante iTunes.....	221
7.2.3 iPhone Backup Extractor.....	223
7.2.4 Acceso a un backup encriptado .....	225
7.2.5 Adquisición lógica con herramientas de terceros.....	226
7.2.6 Adquisición física de un iPhone.....	227
7.2.7 Jailbreaking .....	228
7.2.8 Adquisición basada en técnicas de jailbreaking .....	229
7.2.9 Adquisición de otros dispositivos Apple.....	232

7.3 DISPOSITIVOS ANDROID .....	233
7.3.1 Introducción a Android .....	234
7.3.2 Adquisición de la tarjeta de memoria.....	235
7.3.3 Acceso al terminal Android .....	236
7.3.4 Utilidades de sincronización .....	236
7.3.5 Acceso mediante Android SDK .....	237
7.3.6 Algunas nociones básicas de Android Debug Bridge .....	239
7.3.7 Significado del rooting en Android.....	240
7.3.8 Adquisición física mediante dd .....	242
7.3.9 Examen de la memoria.....	243
7.4 RESTO DE DISPOSITIVOS Y PROCEDIMIENTOS .....	243
7.4.1 Supervivientes .....	243
7.4.2 Adquisición mediante Cellebrite UFED .....	244
7.5 PROCEDIMIENTOS Y RIESGOS .....	245
7.5.1 Alteración de las pruebas .....	245
7.5.2 Recomendaciones ACPO .....	246
7.5.3 Intervención de un dispositivo móvil.....	246
7.5.4 Riesgo legal .....	249
7.5.5 Privacidad.....	249
<b>CAPÍTULO 8. INVESTIGACIÓN DE IMÁGENES DIGITALES .....</b>	<b>251</b>
8.1 INFORMÁTICA FORENSE E IMÁGENES DIGITALES.....	252
8.2 IMÁGENES MANIPULADAS.....	253
8.2.1 ¿Verdadero o falso?.....	253
8.2.2 ¿Cómo funciona una cámara digital?.....	254
8.2.3 Interpolación e inconsistencia estadística .....	256
8.2.4 Artefactos .....	256
8.2.5 Zonas clonadas .....	257
8.2.6 Inconsistencias en la iluminación.....	258
8.2.7 E.L.A. (Error Level Analysis).....	260
8.3 UTILIZACIÓN COMO HERRAMIENTA FORENSE.....	261
8.3.1 La imagen digital como prueba.....	262
8.3.2 Recomendaciones SWGIT .....	263
8.3.3 Buenas prácticas.....	264
8.3.4 Adquisición de imágenes en formato RAW.....	265
8.4 METADATOS GRÁFICOS .....	266
8.4.1 Exif .....	267
8.4.2 IPTC-IIM y Adobe XMP .....	267

8.4.3 Instalación y manejo de Exiftool.....	268
8.4.4 Ejemplo de aplicación .....	269
8.4.5 Limitaciones .....	271
<b>CAPÍTULO 9. HELIX.....</b>	<b>273</b>
9.1 UNA DISTRIBUCIÓN DUAL.....	274
9.1.1 ¿Qué es Helix? .....	274
9.1.2 Características y novedades .....	274
9.1.3 Obtención de Helix.....	275
9.1.4 Arranque en vivo .....	276
9.1.5 CD autoarrancable.....	277
9.2 HELIX SOBRE UN SISTEMA EN FUNCIONAMIENTO .....	278
9.2.1 Interfaz .....	279
9.2.2 Información del sistema .....	280
9.2.3 Examen de la información volátil .....	281
9.2.4 Información de discos .....	281
9.2.5 Información de memoria RAM .....	282
9.3 ADQUISICIÓN DEL SISTEMA EN VIVO .....	283
9.3.1 Orden de volatilidad .....	283
9.3.2 Adquisición de memoria RAM .....	284
9.3.3 Recolección de información volátil .....	285
9.3.4 Imágenes de discos .....	286
9.3.5 Examen de un sistema en funcionamiento .....	288
9.3.6 Helix3 Pro™ Receiver .....	288
9.4 HELIX AUTOARRANCABLE .....	291
9.4.1 Live-CD Linux .....	291
9.4.2 Algunos aspectos de interés forense en Helix .....	292
9.4.3 Helix en una máquina virtual .....	293
<b>CAPÍTULO 10. HERRAMIENTAS SOFTWARE .....</b>	<b>295</b>
10.1 DISTRIBUCIONES LINUX .....	295
10.1.1 Backtrack .....	295
10.1.2 Knoppix .....	297
10.1.3 SystemRescueCD .....	299
10.1.4 CAINE .....	300
10.1.5 Slackware .....	302
10.2 VIRTUALIZACIÓN .....	306
10.2.1 VMware .....	307

10.2.2 VirtualBox.....	308
10.2.3 Listado de herramientas .....	309
<b>CAPÍTULO 11. CONCLUSIONES.....</b>	<b>313</b>
11.1 ESCENARIOS Y APLICACIONES .....	314
11.1.1 En el Juzgado .....	314
11.1.2 Investigaciones en organizaciones y empresas .....	314
11.1.3 Particulares y compañías de seguros.....	315
11.1.4 Sector público y seguridad nacional .....	315
11.2 OBSTÁCULOS .....	316
11.2.1 Destrucción intencionada de la evidencia .....	316
11.2.2 Tecnologías antiforenses.....	318
11.3 DESAFÍOS PARA EL FUTURO .....	319
11.3.1 Clusters.....	319
11.3.2 Cloud computing.....	321
11.4 ALGUNAS RECOMENDACIONES.....	323
11.4.1 La vida no es bella.....	323
11.4.2 Para terminar .....	324
<b>BIBLIOGRAFÍA.....</b>	<b>327</b>
<b>ÍNDICE ALFABÉTICO.....</b>	<b>329</b>

## INTRODUCCIÓN

Apreciada lectora o apreciado lector, sé que está ahí. Puedo verle, de pie frente al anaquel, en busca de algo que supone que necesita, pero no sabe bien lo que es, y tratando de decidir si el ejemplar que tiene entre manos podrá satisfacer sus expectativas o aportarle algo que valga el precio que figura en la contraportada. Tal vez es usted un agente de policía que trabaja en el departamento de delitos tecnológicos del cuerpo, una periodista a la que le han pedido que escriba un artículo sobre el cibercrimen, un estudiante de Ingeniería Informática o, simplemente, alguien interesado por estos temas. Da igual. Si tiene por costumbre empezar a hojear un libro por este apartado, aquí tiene algunas palabras respecto al contenido y a la intención.

Lo que tiene entre manos ha sido escrito desde el propósito de compartir, una filosofía que puede resultar más útil de lo que creemos para remontar las dificultades de una época de crisis en todos los órdenes como la que ahora estamos viviendo: económica, laboral, social y de paradigmas tecnológicos. Lo que el autor pretende es, ni más ni menos, resumir en una obra de poco más de trescientas páginas el conocimiento adquirido a lo largo de varios años en una materia que le apasiona: lecturas, conferencias, seminarios, redacción de artículos en revistas y medios de Internet, cierta actividad docente en cursos y jornadas sobre software libre y cierta experiencia práctica. Haré cuanto esté a mi alcance para transmitirle a usted el mismo entusiasmo. Así mismo, fiel a aquel espíritu del compartir y consciente de que todo lo que tengo es de prestado, asumo el compromiso de no omitir, en el momento y lugar que corresponda, el nombre de los autores y especialistas en la materia cuyo trabajo haya supuesto una inspiración para este libro.

La Informática Forense, según fue definida en el primer DFRWS (Taller de Investigación Digital Forense) celebrado por un grupo de expertos en el año 2001, consiste en el empleo de métodos científicos comprobables para preservar, recolectar, validar, identificar, analizar, interpretar, documentar y presentar evidencias digitales procedentes de fuentes digitales con el propósito de hacer posible la reconstrucción de hechos considerados delictivos o ayudar a la prevención de actos no autorizados y capaces de provocar una alteración en operaciones planificadas de organismos y empresas. Un estudio sistemático de toda esta constelación de elementos, guiado por buenas prácticas y estándares reconocidos, es imprescindible no solo para llevar a cabo una investigación eficaz, sino para asegurar la validez jurídica de las evidencias recuperadas, de cara a su utilización posterior en tribunales, autoridades públicas y departamentos de seguridad de las grandes empresas.

La presente obra pretende transmitir al lector nociones inevitablemente breves y superficiales de una materia que pese a lo específico de su denominación posee carácter multidisciplinar y abarca áreas especializadas de enorme complejidad, desde tecnologías de almacenamiento a bajo nivel hasta bases de datos, pasando por sistemas de archivos, sistemas operativos, tecnología de redes, legislación, hacking informático, ingeniería social, psicología y otros ámbitos diversos. Se pondrá énfasis en temas clave como los requerimientos de la investigación forense y la gestión adecuada de evidencias digitales. Se hablará de herramientas tanto propietarias como de código libre y de los desafíos esperables como consecuencia del progreso acelerado de las tecnologías de la información.

Debido a la falta de espacio no será poco lo que haya que omitir. El objetivo que el autor se ha propuesto consiste en proporcionar al lector un punto de partida a partir del cual pueda proseguir mediante su propia iniciativa. A tal fin se le facilitan recursos bibliográficos y enlaces de Internet. Seguramente el lector, si ya tiene conocimientos de Informática Forense, comprobará que no está incluido todo lo que a su juicio debiera, en virtud de su experiencia propia o en comparación con las obras de referencia sobre esta materia escritas en otros idiomas, principalmente en inglés. Si hay algo que echa en falta, ello se debe, además de a la impericia del autor, que se considera experto no por lo que sabe sino por lo que cada día tiene ocasión de aprender, a la inseguridad de si tal especialidad o tema sigue siendo importante en relación con otros más perentorios o novedosos, y también al deseo consciente de poner énfasis sobre aspectos y tendencias relevantes en el panorama actual de la investigación informática forense, que al igual que el de las tecnologías de la información se encuentra sometido a las fuerzas incontenibles del cambio. De modo que si el lector no está dispuesto a perdonar que el autor haya omitido el análisis forense del iPaq o el Palm V, o cualquier otro de sus dispositivos predilectos, al menos debe entender que lo

antiguo ha de ceder paso a lo nuevo, y que esas páginas en blanco las necesitaban con urgencia otros temas como el análisis forense de *smartphones* o la investigación en entornos de *cloud computing*.

Por razones similares se ha querido desde un primer momento poner más énfasis en los procesos que en el empleo de herramientas específicas. El lector aprovechará mejor su tiempo aprendiendo el funcionamiento subyacente de los sistemas que adquiriendo destreza en comandos y programas que le permitan lograr unos resultados específicos sin tener la menor idea de cómo funcionan en realidad. Una vez comprendidos los procesos se verá más libre a la hora de seleccionar el software que necesita o —si tiene experiencia como programador— incluso escribir sus propias herramientas. También podrá combinar las ya existentes en forma de programas desarrollados en lenguajes de alto nivel o mediante *scripts* de Perl, Python, Ruby o similares. Las herramientas irán siendo presentadas a medida que se discutan los temas correspondientes —adquisición de soportes, análisis de sistemas de archivos, investigación del Registro de Windows, historiales de Internet, correo electrónico, etc.—, con explicaciones relativas a instalación, uso y otras particularidades.

En este libro se emplean herramientas de código libre para Linux con un doble propósito: por un lado económico —pocos usuarios están en situación de financiarse el software utilizado por los departamentos de policía y las grandes empresas— y en segundo lugar pedagógico. Naturalmente se hará referencia al software comercial porque se trata de productos prestigiosos de gran calidad que marcan la pauta en este campo (EnCase de Guidance, FTK, SMART). Sin embargo, y por motivos de asequibilidad, las explicaciones más detalladas corresponden a herramientas de código libre: The Sleuth Kit, Pasco, Galleta, Testdisk, Scalpel, etc. Se trata en su mayor parte de utilidades en línea de comando que se ejecutan en una consola bash o el entorno de Unix Cygnus para Windows. La línea de comando exige un esfuerzo de aprendizaje que por razones fáciles de comprender puede resultar más arduo para aquellos lectores que por haber nacido con posterioridad al año 1980 no tuvieron que aprender a manejar ordenadores con Unix o MSDOS y lo único que han conocido son entornos gráficos. Pero tiene la ventaja de una proximidad inmediata al plano de operaciones informáticas de bajo nivel —ejecución de código en forma de órdenes directas y scripts para manipulación de datos— en el cual se desarrolla gran parte de la actividad Informática Forense. También permite al usuario comprender los procesos subyacentes y los métodos básicos de recuperación de datos. Así mismo el software libre, aparte de las ya conocidas ventajas de coste, ofrece beneficios educativos de indudable interés. Hasta hace pocos años los dos únicos itinerarios curriculares para formarse como especialista en Informática Forense pasaban por las Fuerzas Armadas o la Policía. Gracias al software libre la profesión se

encuentra ahora al alcance de colectivos más amplios y centros de enseñanza que se planteen organizar ciclos formativos sobre la materia sin costes adicionales y aprovechando la infraestructura existente.

No pretendo disimular ante el lector la presencia de numerosos defectos de los que se irá dando cuenta durante su avance a lo largo de la obra. Uno de los que quizás le llame más la atención es la ausencia clamorosa del entorno Apple —excepto de esta omisión las explicaciones relativas al iPhone y otros aparatos provistos de OSX y sistemas de archivos HFS+ en el capítulo sobre investigación forense de dispositivos móviles—. Haber prescindido de productos y tecnologías Apple no tiene nada que ver con las preferencias del autor y mucho menos con el propósito de quitarle relevancia al tema. Muy al contrario, pese a la baja probabilidad de que un investigador se las tenga que haber en el transcurso de su trabajo habitual con ordenadores OSX o MacIntosh, Apple como plataforma dispone de productos propietarios de gran potencia para el análisis forense, además de las ventajas de una tecnología informática que va cinco años por delante de su tiempo. Simplemente no hay espacio para tratar el tema con cierta profundidad. En otra parte el lector podrá hallar gran cantidad de información sobre investigación forense de sistemas OSX y Apple en forma de libros, páginas web y artículos en revistas especializadas.

Libros y páginas web, sin embargo, no son a su vez más que otros puntos de partida. La verdadera experiencia procede del trabajo personal y un seguimiento constante de los avances y las noticias de actualidad en el campo de la Informática Forense. Lo más que puedo hacer para alimentar el entusiasmo del lector, aparte de haber escrito acerca de sistemas de archivos, cadenas de custodia y cómo recuperar archivos borrados, es llamar su atención sobre la importancia de un hecho histórico irrefutable: las tecnologías digitales están cambiando el mundo en mayor medida y profundidad que cualquier otro avance realizado por el ser humano.

El poder de las tecnologías digitales constituye desde hace años un tópico habitual en los medios. Sorprendentemente, y a diferencia de tiempos pasados, este poder no es privilegio de unos pocos, sino que está al alcance de una parte considerable de la población del globo. En el mundo en que vivimos hay más microprocesadores que seres humanos. Hace tiempo que el ordenador dejó de ser una mercancía de lujo para convertirse en un artículo polivalente, ubicuo e irrenunciable en la actual cultura popular de masas: los ordenadores portátiles y de sobremesa son nuestras herramientas de trabajo, nuestro canal de comunicaciones, nuestra fuente de diversión. Tratándose de algo omnipresente, con una potencia y unas posibilidades que los usuarios aprovechan tan solo en una medida irrisoria, quizás ni siquiera hará falta explicar por qué la Informática Forense se ha convertido en una disciplina importante y cada vez más demandada.

Los dispositivos informáticos –ordenadores, soportes digitales, agendas electrónicas, teléfonos móviles, *routers*, etc.– son protagonistas de una cantidad creciente de actividades ilícitas, tanto en condición de cuerpo del delito como de herramientas utilizadas en la comisión de aquellas. La lista de ejemplos es abrumadora y todos los días los medios de comunicación suman algo nuevo a ella: espionaje industrial, pornografía infantil, piratería de producto, empleados desleales, fraudes de tarjetas de crédito, acoso moral, correo electrónico no deseado, ataques de denegación de servicio y un larguísimo etcétera. La policía hace lo que puede, el trabajo se va acumulando en grandes colas dentro de los laboratorios y urge la formación de un creciente número de especialistas.

La diferencia entre lo analógico y lo digital va más allá de las ventajas que se podrían deducir de una simple explicación académica. Ya sabe, aquello de que una señal analógica es aquella que adquiere valores dentro de un rango continuo, generalmente como respuesta proporcional a una magnitud que varía de modo también continuo, mientras que los impulsos digitales implican la representación de estados a partir de un conjunto de valores discretos. Rara vez nos detenemos a pensar en lo que este cambio de paradigma trae consigo realmente. Las tecnologías analógicas –radio, televisión, radar, sensores electrónicos, etc.– nos permiten obtener representaciones del mundo real: fotografías, mapas, registros sísmicos, radiografías, películas documentales, discos de música clásica. Las tecnologías digitales sirven para lo mismo, y además nos permiten interactuar con la realidad, modificarla e incluso reinventarla de formas tan espectaculares que en la práctica los límites vienen dados únicamente por la imaginación. Podemos restaurar películas antiguas, recuperar la voz natural de Caruso después de haber suprimido la distorsión de la bocina a través de la cual se grabó, corregir una fotografía desenfocada, restaurar un manuscrito griego reutilizado por un copista medieval y saber si un cuadro de Rubens es auténtico. También podemos crear paisajes y arquitecturas inexistentes, dar vida a ogros verdes y alienígenas juerguistas, localizar el piso franco de una célula terrorista cruzando bases de datos o segmentar un mercado hasta el último consumidor.

Estas posibilidades también están al alcance de los delincuentes. Este no solo es un buen argumento para fomentar la formación en tecnologías informáticas forenses dentro de las organizaciones encargadas de velar por la seguridad y la justicia, sino que los propios hechos y las necesidades tácticas y estratégicas de la lucha contra el crimen conducen a ello de modo inevitable. Policía, Justicia, administradores de redes, responsables de seguridad de las grandes empresas y miembros de la clase política se ven abocados a un esfuerzo permanente de aprendizaje y actualización que en ningún momento podrán descuidar si no quieren verse superados por las fuerzas disgregadoras de la sociedad y los enemigos del orden público que se sirven de las mismas tecnologías para sus fines ilícitos.

## Capítulo 1

# SE HA ESCRITO UN CRIMEN DIGITAL

### 1.1 UN DÍA COMO OTRO CUALQUIERA

Son las 7:15 de lo que se anuncia como una mañana cálida y luminosa de junio. Usted está sentada frente al ordenador, con una taza de café humeante sobre el escritorio, mientras su marido riega las macetas del balcón y su hija mayor prepara el desayuno en la cocina. Lo mismo de cada día: consultar el correo electrónico y leer los titulares de prensa. Entonces descubre dos mensajes extraños que el sistema antispam de su proveedor de Internet no ha sido capaz de filtrar. Uno de ellos es una carta nigeriana: una tal Sra. Kathleen Tsimenga le propone compartir veinte millones de dólares depositados en Suiza por un dictador africano a cambio de facilitarle un número de cuenta para la operación. En el otro mensaje, la Agencia Tributaria insiste en que le ha cobrado de más y le pide sus datos bancarios (contraseña incluida) para devolverle el dinero. ¿Qué hacer? Muy fácil: a la papelera con los dos; luego sale de su correo, cierra el sistema y acude a reunirse con los suyos para disfrutar del único momento del día en que pueden estar juntos y pasar un buen rato en familia antes de salir para el trabajo.

Usted piensa que si alguien tiene problemas con el ordenador es porque se lo ha buscado. Y no le falta su parte de razón. Su ordenador, recién traído de la tienda, con Windows 7, configuración de mínimos, todas las defensas levantadas y unas buenas herramientas de control parental, es de los que predicen con el ejemplo. Hasta la fecha usted no ha sufrido ataques de troyanos ni *phishing*. Tampoco ha perdido archivos ni detectado intrusiones de los estudiantes que alquilan el piso de al lado. En su percepción, todos los males de la sociedad de la

información parecen limitarse a eso, a travesuras de adolescentes y anomalías que de vez en cuando obligan a llamar al servicio técnico.

Por desgracia el mundo no es tan sano como cree. En esos momentos la vecina del primero izquierda está pasando por un trance bochornoso. Anoche la Policía Nacional halló en el disco duro del ordenador una cantidad significativa de imágenes de sexo con menores de edad. Ante la perplejidad y el estupor de su familia, la señora de la casa –no se sorprenda: también las mujeres trafican con pornografía infantil–, no pudo seguir soportando la presión y confesó. Más abajo, en una lonja de comestibles del mismo bloque, una banda de delincuentes ha montado un fraude de venta de ordenadores portátiles por Internet en el que ya han caído más de noventa personas. Los estafadores ignoran que la Guardia Civil les sigue el rastro desde hace días y en estos momentos está tomando posiciones para entrar en el local. Usted tampoco sospecha que sus vecinos estudiantes disponen de acceso gratis a Internet tras haber craqueado la mitad de los *routers* inalámbricos del vecindario. También furgonean en los recursos compartidos de varios ordenadores, incluyendo el portátil de su hija, en busca de documentos y archivos personales: fotos digitales, correo electrónico Outlook, declaraciones de la renta y correspondencia personal.

Ya conoce el dicho: ojos que no ven... Pero estas cosas no solo suceden en su portal. Usted trabaja como ejecutiva en una empresa del sector de Defensa. Cuando por las tardes sale de su despacho, cambiando saludos con las mujeres de la limpieza que comienzan su turno, ¿cómo podría imaginar que una de ellas posee conocimientos de informática suficientes para arrancar ordenadores con un Live-CD Linux, abrir una consola bash, teclear comandos que permiten el montaje de particiones en modo lectura y trasladar archivos confidenciales del departamento que usted dirige a una llave USB, sin que en los *logs* del sistema quede el menor rastro de la operación? No tenemos ni idea de para quién trabaja: quizás para la competencia, para algún ejecutivo desleal de la misma empresa o –peor aún– para un gobierno extranjero.

Mientras introduce la llave en el ascensor del garaje, tras haberse despedido de los suyos, la Guardia Civil ya está levantando el portón de la lonja. A lo lejos se oyen ruidos metálicos y una caja de herramientas cayendo al suelo. Usted piensa que se trata de unos operarios acarreando material para una obra. Todo sucede tan deprisa que únicamente se enterará de lo que pasó al día siguiente cuando lea la noticia en el periódico.

Hora y media después, en el Grupo de Delitos Tecnológicos, un agente firma el acuse de recibo del material incautado –dos iPhones, tres ordenadores de sobremesa, un portátil, gran cantidad de llaves USB, DVD y unas cuantas tarjetas de crédito– y piensa con resignación en la cola de trabajo que se acumula en el

laboratorio. Cuando la primera unidad de investigación forense comenzó a funcionar hace algunos años, podían permitirse el lujo de escudriñar hasta el último byte de cada soporte de datos que les llegaba. Luego se produjo un auténtico alud de material probatorio y hubo que asignar una cuota de tiempo a cada caso. Al principio 17 horas, ahora 15, en el futuro probablemente menos. El funcionario pone manos a la obra y descuelga el teléfono para avisar a los otros miembros del equipo.

## 1.2 INTERVENCIÓN

Lo que acaba de leer no es el comienzo de un reportaje sensacionalista, sino una instantánea de la realidad. La investigación forense de medios digitales comienza aquí, desarrollándose a lo largo de una serie de etapas. En el Estado de Derecho la tecnología no importa tanto como la forma de hacer las cosas, aplicando buenas prácticas que aseguren el cumplimiento estricto de la ley y preserven el valor jurídico de las evidencias recolectadas.

### 1.2.1 Los primeros en llegar

En el ámbito cultural anglosajón la figura del *first responder* o primer intervintante, corresponde a la persona que llega al escenario de los hechos antes de que nadie haya tocado nada. Su cometido consiste en asegurar la evidencia y garantizar la cadena de custodia con vistas al empleo posterior de las pruebas en un proceso judicial. Normalmente se trata de un agente de policía equipado con una cámara digital, rotulador para CD y un juego completo de destornilladores. Si hay un ordenador encendido, no permitirá que nadie lo toque. Lo primero que hará es fotografiar las pantallas de los ordenadores. También fotografiará los equipos y todo el material.

### 1.2.2 Apagado sin más

Acto seguido apagará los equipos sacando el enchufe de la red. Esto normalmente no se debe hacer nunca, pero la intervención del *first responder* obedece a circunstancias especiales. Un apagado ordenado podría eliminar parte de los archivos temporales, que por lo general se borran automáticamente al detener el sistema, perdiéndose así elementos potenciales de evidencia o quedando alterada la prueba del delito. El apagado de equipos por la vía explícita de extraer un enchufe de la red o cortar el sistema de alimentación ininterrumpida ayuda a evitar que se formulen posteriormente conjeturas relativas a la integridad de la cadena de custodia. A veces el apagado no es inmediato sino que antes de quitar el enchufe de la red un experto en Informática Forense, valiéndose de una herramienta de software adecuada, extrae una imagen de la memoria del ordenador para analizar

los procesos en ejecución. En todo momento ha de evitarse todo aquello que pueda iniciar un autoapagado del sistema.

### 1.2.3 Objetos intervenidos

Para el investigador forense son importantes los elementos y objetos que se mencionan a continuación:

- Ordenadores de sobremesa, portátiles, agendas electrónicas, smartphones y teléfonos móviles.
- Discos duros de ordenador.
- Discos duros extraíbles y portátiles.
- Unidades de almacenamiento externas de todo tipo (llaves USB, reproductores MP3, tarjetas para cámara digital, etc.).
- Discos ópticos CD y DVD grabables y regrabables.
- Disquetes.
- En general, todo soporte externo que sirva para un almacenamiento de datos permanente.

Por el contrario carecen de interés los elementos siguientes, a no ser que constituyan objeto de alguna actividad delictiva o estén relacionados con la investigación en otros aspectos de la investigación forense:

- Monitores de ordenador y pantallas TFT (a no ser que muestren algo interesante, en cuyo caso deberán ser fotografiadas con una cámara digital).
- Teclados y ratones.
- Impresoras.
- Cables.
- CD/DVD-ROM grabados en fábrica.
- En general todos aquellos dispositivos que no sirvan para almacenar datos.

En lo que respecta a tales objetos basta con inventariarlos y fotografiarlos, indicando en el informe lugar, momento y circunstancias de su hallazgo. El análisis de estos elementos raramente aporta claridad a la investigación forense, salvo en el caso de las impresoras y en la circunstancia de que las mismas lleven tarjetas de

memoria insertadas en alguna ranura, chips de RAM no volátil en su interior, o tengan defectos que permitan relacionar una copia en papel con la impresora utilizada para obtenerla.

Existen otros elementos de difícil clasificación cuyo interés depende de las circunstancias, y que en determinadas ocasiones pueden resultar útiles como objetos de prueba: grabadores de bandas magnéticas (utilizados en la manipulación de tarjetas), lotes de tarjetas SIM para teléfonos móviles, *scanners* de radiofrecuencia, grabadoras de audio, copiadoras de discos, etc. Aunque el informático forense quizás no pueda hacer gran cosa con este tipo de materiales, quizás resulten útiles para un investigador criminológico.

### 1.3 LA AMENAZA DIGITAL

Decir que en la mayor parte de los países el delito informático es desde hace años un fenómeno al alza quizás no suponga nada nuevo para el lector. Eche un vistazo a la tabla 1.1. Se trata de un resumen de los principales tipos de acciones delictivas relacionadas con la informática que se registraron en Alemania durante el año 2010, junto con la proporción de casos resueltos y las cifras comparables correspondientes al año anterior. Basta un somero examen de estos datos para advertir dos tendencias significativas: el progresivo incremento en el número de la mayor parte de modalidades delictivas de un año para otro —que dicho sea de paso es continuación de ejercicios anteriores— y la paralela disminución en el número de casos resueltos. Podemos asumir que si bien de unos países a otros puede haber diferencias significativas en cuanto a cifras, casos de resolución y distribución entre los diversos tipos de acciones criminales, la tendencia al desplazamiento sostenido del equilibrio a favor de los transgresores de la ley y en contra de las fuerzas del orden es característica de la mayor parte de las sociedades industrializadas, incluyendo a España.

Modalidad de delito	Casos denunciados		Variación		Resueltos %	
	2010	2009	Absoluta	%	2010	2009
Delitos informáticos	84.377	74.911	9.466	12,6	35,8	37,5
Desglose:						
Estafas mediante tarjetas de débito adquiridas ilegalmente	23.612	23.163	449	1,9	40,7	38,5
Estafas por ordenador	27.292	22.963	4.329	18,9	30,2	34,8
Estafas relacionadas con el uso ilícito de telecomunicaciones	7.993	7.205	788	10,9	44,0	41,1

Falsificación de datos y procesamiento ilícito de los mismos	6.840	6.319	521	8,2	52,0	53,2
Alteración de datos y sabotaje	2.524	2.276	248	10,9	32,1	36,9
Espionaje y adquisición no autorizada de datos	15.190	11.491	3.699	32,2	24,0	22,4
Piratería de software (para uso particular)	794	1.351	-557	-41,2	94,1	96,7
Piratería de software (con intenciones lucrativas)	132	143	-11	-7,7	97,0	95,8

Tabla 1.1. Estadísticas de la criminalidad informática en Alemania  
(Fuente: Bundeskriminalamt, Computerkriminalität – Berichtsjahr 2010)

Las estadísticas no son útiles solamente por el cuadro que nos pintan del pasado, sino por lo que dicen acerca de la posibilidad de una repetición de actos de naturaleza similar en el futuro. En los tiempos que vivimos, y teniendo en cuenta las cifras correspondientes a ejercicios anteriores, resulta fácil hacerse una idea del tipo de panorama al cual nos conduce una simple extrapolación. Antes de proseguir es necesario tratar de la cuestión de hasta qué punto ordenadores y redes se hallan expuestos al peligro de un ataque. ¿Cómo de probable es una intrusión en mi sistema? De llegar a darse, ¿qué daños puede suponer? ¿Quiénes son los atacantes potenciales? ¿Qué persiguen? Pero en primer lugar, y sobre todo, ¿qué es el delito digital?

#### 1.3.1 El delito informático

El delito informático en sentido estricto, con su tipificación exacta en el Código Penal, es algo que no existe. Su propio concepto resulta polémico para los juristas. En lo único que se está de acuerdo es en el hecho de denominar bajo ese término aquellas acciones delictivas, bien de tipo tradicional bien novedosas, que se cometen a través de un nuevo medio —las redes informáticas— o con los recursos facilitados por aquellas: ordenadores y herramientas de software.

Inevitablemente los actos ilegales vinculados a las tecnologías de la información requieren el uso de métodos de investigación informáticos. El Consejo de Europa, con motivo de un Convenio sobre Ciberdelincuencia celebrado en 2001, estableció unas categorías que desde entonces gozan de una amplia aceptación, dividiendo los delitos informáticos en cuatro grandes grupos:

- Delitos contra la confidencialidad, la integridad y la disponibilidad de los datos y sistemas informáticos. Ejemplos típicos dentro de esta clase de actuaciones ilícitas son la intercepción de datos, el acceso no

- autorizado a redes y sistemas, interferencias en el funcionamiento de instalaciones y el empleo de aparatos que hagan posible todo lo anterior.
- Delitos cuyo propósito sea la falsificación, supresión o borrado de datos.
  - Delitos relacionados con contenidos: el más grave de todos ellos es la pornografía infantil.
  - Delitos relacionados con la vulneración de derechos de propiedad intelectual.

Sin embargo en el transcurso de unos pocos años el avance de la tecnología, el crecimiento de Internet y otros factores culturales, sociales y geopolíticos han dejado obsoleta la clasificación anterior. Actualmente se cometen actos ilícitos como acoso moral y ciberterrorismo que no encajan en las categorías establecidas por el Consejo de Europa y que son claramente delictivas.

En España, que ratificó el Convenio con efecto al 1 de octubre de 2010, la práctica habitual consiste en aplicar las tipificaciones delictivas del Código Penal más próximas a las categorías establecidas por la autoridad europea (delitos de corrupción y prostitución de menores, delitos de descubrimiento y revelación de secretos, relativos al mercado y los consumidores, daños, estafas, etc.), así como otros actos ilícitos vinculados (abusos sexuales, amenazas, injurias, consumo fraudulento de energía eléctrica). Lo propio se aplica en cuanto a la determinación de las responsabilidades civiles derivadas de las conductas delictivas.

Fuera del ámbito penal el ordenamiento jurídico español se amplía con normas legislativas de reciente incorporación, como por ejemplo la Ley de Servicios para la Sociedad de la Información y el Correo Electrónico, la Ley de Protección de Datos, la Ley General de Telecomunicaciones, la Ley de Propiedad Intelectual y la Ley de Firma Electrónica.

### 1.3.2 Evaluación del riesgo

La probabilidad de que una amenaza digital llegue a producirse depende de varios factores: (a) frecuencia de la misma según la estadística y la experiencia histórica; (b) vulnerabilidad de los sistemas atacados; (c) valor de los datos; (d) motivación del atacante y (e) relevancia mediática de la entidad atacada.

En la práctica resulta posible evaluar el riesgo en función de dos factores principales: en primer lugar la misma probabilidad del hecho hostil –intrusión en sistemas, sabotaje, robo de datos– y, segundo, el valor del bien comprometido –información confidencial, documentos de clientes o proveedores, datos críticos

para el funcionamiento de los sistemas, etc.–. Sería aconsejable que con carácter preventivo se lleve a cabo un análisis de riesgos centrado en la probabilidad de los ataques y una evaluación de los daños económicos asociados a los mismos, tomando a renglón seguido las medidas técnicas, organizativas, legales o de infraestructura que se consideren oportunas. En cualquier caso, una vez que las actuaciones ilícitas han tenido lugar, tales procedimientos resultan inexcusables si se desea evitar que los incidentes vuelvan a producirse en el futuro.

Durante los últimos años la escena del delito digital ha experimentado un profundo cambio, pasando de ser un entorno en el que de vez en cuando se cometían hechos aislados a convertirse en un fenómeno de masas, con millares de delincuentes, activistas políticos, miembros de bandas organizadas y simples gamberros. La culpa la tienen por un lado el monocultivo de sistemas operativos y aplicaciones, y por otro la explosión de Internet con su enorme variedad de servicios, la estandarización de sus protocolos y la presencia inevitable de fallos de configuración y diseño en el software. Todo ello ha conducido a una multiplicación exponencial de los incidentes de seguridad.

### 1.3.3 Los motivos del agresor

En los comienzos de la revolución informática los ataques obedecían a dos motivaciones básicas: rebeldía y codicia. Todas las modalidades actuales del delito informático, independientemente de cómo se perpetren y del juicio moral que nos merezcan, son variaciones de esos dos temas fundamentales. Las personas que llevan a cabo una intrusión en un sistema lo hacen por alguna de estas razones concretas: necesidad de destacar socialmente, activismo político, curiosidad, deseo de obtener ganancias económicas o porque un gobierno extranjero las haya contratado para sustraer información o cometer actos de sabotaje. Si bien al principio el ansia de notoriedad era el incentivo más poderoso, en una época en la que individuos aislados o pertenecientes a pequeños grupos exhibían un comportamiento antisocial para llamar la atención o cosechar méritos que les permitieran ser admitidos en alguna comunidad prestigiosa de *hackers*, en la actualidad la tendencia apunta al predominio del delito informático con motivaciones clara y exclusivamente económicas. He aquí algunos ejemplos característicos:

- Ataques contra infraestructuras de red y sitios web.
- Robo de datos de cuentas bancarias mediante técnicas de *phishing*.
- Envíos de correo no solicitado (*spam*).

- Fraudes en sitios de compras *on line*.
- Lesión de derechos de propiedad intelectual.
- Chantaje, acoso moral y extorsión a través de las redes sociales.
- Implantación de troyanos y programas parásitos.
- Robo de contraseñas.
- Gestión de una *botnet*.
- Pornografía infantil.

Términos anticuados como *hacker*, *cracker*, *phreaker*, *ciberpunk*, *script kiddie*, etc., son el legado de una época en que adolescentes inadaptados se entretenían colándose en los sistemas gubernamentales y de las grandes empresas en busca de errores de configuración del software. Ocasionalmente, y según la calidad del guión, estos individuos caían en manos de algún malhechor que los utilizaba para un golpe de gran envergadura, aventuras de espionaje o simplemente desencadenar una guerra nuclear. Esta visión, mitificada por una literatura popular y un cine realmente muy poco serios, cede sin rechistar ante una realidad mucho más prosaica caracterizada por el predominio de las dos grandes amenazas que traen de cabeza a los expertos en seguridad: la delincuencia organizada y la ciberguerra, con una gama de actividades que van desde el espionaje y las intrusiones en ordenadores de organismos oficiales hasta el intento de dañar infraestructuras críticas. La triste realidad es que en el ciberespacio quienes dictan la ley no son los héroes del ciberpunk, ni siquiera la CIA, y mucho menos Bill Gates, sino las bandas organizadas. Este panorama es el que con toda probabilidad va a marcar la pauta durante los próximos años.

### 1.3.4 Amenazas internas y externas

Un ataque puede proceder de diferentes lugares. Antiguamente venían desde algún terminal situado en el interior de la organización, pero el predominio de las redes ha propiciado en los últimos años un desplazamiento de la amenaza a lugares remotos, principalmente a través de Internet. Con frecuencia basta un ordenador, una conexión de banda ancha y algunos conocimientos técnicos para planificar la ofensiva. Sin embargo, pese al incremento en el número de intrusiones externas, la mayor parte de los estudios estadísticos coinciden en el carácter aún predominante de la amenaza interna. Esto se explica por un número de causas. En primer lugar la precariedad de mecanismos de defensa en el interior de las redes

facilita el abuso de sistemas y el robo de datos por medio de actos que en la mayor parte de los casos no trascienden al exterior por razones obvias. Deslealtad, espionaje empresarial, revelación de secretos, obtención de información privilegiada, intentos de extorsión o, en algunos casos, el simple deseo de venganza, aparte de muy difíciles de demostrar, no son temas populares, al menos dentro del perímetro defensivo de la empresa. Con frecuencia la dirección prefiere silenciarlos antes que darles una publicidad que puede comprometer el prestigio de la casa o perjudicar el posicionamiento de su marca en los mercados.

El agresor interno no solo cuenta con mayores facilidades de acceso a la información, sino con un conocimiento detallado de los procesos empresariales. También sabe dónde están guardados los datos más valiosos. La problemática de los ataques internos se complica al afectar a un círculo más amplio de personas sin relación orgánica con la empresa pero con acceso a su red: asociados comerciales, proveedores, servicios de auditoría y certificación, clientes incluso. Todo estudio sobre amenazas internas resulta problemático y de ahí que resulte difícil realizar afirmaciones válidas. Las empresas –sobre todo los bancos– temen por los daños que estos incidentes puedan causar a su imagen. La existencia de conductas desleales mina la moral interna de las organizaciones y pone de manifiesto la falta de una cultura empresarial sana. Un empleado que manipula cuentas bancarias o roba los datos de la empresa es como el albañil de aquel chiste soviético que con el pretexto de llevarse un poco de arena para el jardín sustraía carretillas de la fábrica. Mucho más perjudicial que eso incluso, ya que en la economía del conocimiento los procesos empresariales están estrechamente vinculados a operaciones que se llevan a cabo con la ayuda de sistemas de información –transferencias bancarias, gestión de stocks, documentos, control de maquinaria, etc.–. El robo de un archivo informático a veces puede causar más daño que el siniestro de un tren de laminación.

### 1.4 DINÁMICA DE UNA INTRUSIÓN

El ataque informático puede ser de muchos tipos. De hecho, tantos como tipificaciones de delitos determinados existan en el Código Penal y requieran de un ordenador para cometerlos. A veces una banda de ciberterroristas paraliza miles de máquinas o la página web de alguna organización importante. En otro caso un empleado vengativo deja la red caída saboteando un *router*, saca un disco duro del servidor o se lleva las bases de datos de la empresa en una llave USB. La intervención del investigador o las medidas a tomar dependerán de las circunstancias concretas.

Mucho antes de sufrir alguno de estos percances, un ejército anónimo y descoordinado compuesto por miles de aficionados o *script kiddies* habrá probado fortuna atacando a la empresa con herramientas descargadas de Internet cuyo funcionamiento no entienden, pero que siguiendo instrucciones halladas en algún foro de *hackers* ejecutan en sus ordenadores contra cualquier objetivo remoto por ánimo de fastidiar o simplemente para ver qué pasa. Se trata de novatos oportunistas sin auténticos conocimientos de informática. Por su parte un atacante profesional que se haya propuesto abrirse camino hasta el interior de una organización no actuará de manera aleatoria, sino siguiendo una estrategia planificada. Todo intento de intrusión en un sistema informático, bien a través de la misma red local bien desde ubicaciones externas, se desarrolla de acuerdo con un esquema bien definido. El informático forense debe comprender este *modus operandi* para verificar la existencia de la intrusión y elaborar una línea de tiempo que permita esclarecer los hechos.

#### 1.4.1 Footprinting

Antes de llevar a cabo su agresión, el atacante intentará establecer de modo sólido los fundamentos de aquella. En otras palabras: decidirá qué objetivos son los que le interesan. Puede tratarse de un *host* concreto, de un rango de direcciones IP o de un interfaz de telecomunicaciones. Depende de lo que el intruso haya venido a buscar. El resultado de esta fase es una lista con direcciones IP correspondientes al objetivo.

Esta fase de búsqueda de información es silenciosa y disimulada. El atacante se sirve de motores de búsqueda (Google) y portales especializados como Shodan, y sus movimientos se pierden en el ruido de fondo de Internet. Sin embargo en ordenadores incautados existen formas de saber si el usuario ha estado llevando a cabo operaciones de *footprinting*, lo cual sería el caso si en los historiales de comandos y el navegador de Internet aparecen indicios de las actividades siguientes:

- Búsquedas en Whois.
- Rastros de cualquier tipo de investigación o análisis sobre arquitectura de redes (posicionamiento de *routers*, ubicación de rutas *up-* y *downstream* de las organizaciones correspondientes, etc.).
- Consultas a servidores DNS.
- Utilización de herramientas de red como ping, fping o traceroute.

- Determinación de *hosts* activos mediante barridos ping dentro de un rango de direcciones IP.
- Búsquedas de direcciones IP y servidores Web en Shodan.
- Utilización de operadores avanzados de Google en los que como términos de búsqueda figuren mensajes de error correspondientes a determinadas aplicaciones defectuosas.

#### 1.4.2 Escaneo de puertos y protocolos

La conexión de todos los ordenadores del mundo mediante un sistema de direcciones universal constituye sin lugar a dudas uno de los mayores logros de ingeniería de todos los tiempos. Pero no serviría de mucho si cada vez que quisiéramos utilizar el correo electrónico o descargar un archivo por FTP tuviésemos que dejar interrumpida la conexión del navegador de Internet con un sitio web. El manejo de aplicaciones concurrentes a través de direcciones IP compartidas resulta posible gracias a un sistema de números internos llamados puertos similar a un casillero de apartados de correos. Algunos de estos puertos son de uso estándar y se utilizan para el funcionamiento de servicios como correo electrónico (25), navegación web (80), administración remota vía Telnet (21) o comunicación entre sistemas operativos que soportan el protocolo NetBios de Microsoft (139).

Los protocolos son especificaciones de reglas y procedimientos utilizados en todo proceso de comunicación y necesarios para hacer posible el traslado de datos entre ordenadores. Cada servicio se sirve de sus correspondientes protocolos con unas posibilidades de configuración más limitadas que en el caso de los puertos. Podríamos poner un servidor de páginas web en otro puerto que no fuera el 80, pero para hacerlo funcionar necesitamos de cualquier modo HTTP. Así mismo el enrutamiento se lleva a cabo a través del protocolo IP; el envío de correo saliente utiliza SMTP, la entrega de páginas web TCP, y UDP es para servicios de comunicación menos exigentes sin necesidad de acuse de recibo como VoIP o vídeo bajo demanda. Existen numerosos protocolos, quizás más de un millar, algunos para control y administración de redes como ICMP. Todos ellos funcionan formando una pila que más o menos sigue el modelo de capas OSI para redes informáticas. En este modelo, cada capa se encarga de cumplir una misión característica: recoger impulsos eléctricos del cable o del medio de transmisión y agruparlos en tramas, interceptar tramas en función de las máquinas que las envían o reciben, enrutar paquetes entre diferentes ordenadores y redes, gestionar conexiones a través de puertos, etc.

Los puertos constituyen el punto débil más importante de las redes, ya que proporcionan una vía de acceso al interior de los sistemas. En la fase de escaneo de puertos, el atacante, utilizando herramientas de software automatizadas, intenta conectar con los puertos de un *host* para averiguar cuáles están abiertos.

### 1.4.3 Enumeración

Durante la fase de enumeración el atacante intenta averiguar qué aplicaciones funcionan conectadas a los puertos escaneados en las direcciones IP de destino: sistemas operativos, servidores web o herramientas administrativas cuyos defectos de configuración o fallos de software figuren mencionados en alguna lista de *bugs* en Internet. En ocasiones las propias empresas colaboran sin saberlo con el agresor facilitando esta información mediante ofertas de empleo (al plantear perfiles de requisitos TIC), foros de atención al cliente, historias de éxito o simplemente –y aunque parezca increíble, pero es de lo más cierto– publicando diagramas detallados de sus arquitecturas de red.

Al atacante no solo le interesa el tipo de software, sino también las versiones y los parches de seguridad instalados, puesto que de esta información podrá extraer conclusiones relativas a las deficiencias de seguridad explotables con vistas a un intento de intrusión.

### 1.4.4 Penetración y despliegue de exploits

Hemos llegado al punto en el que el atacante, en el caso de persistir en su empeño, estaría traspasando ya los límites de la legalidad. Todo lo que ha hecho hasta el momento es en la mayor parte de los países perfectamente lícito. A partir de aquí comienza a incurrir en responsabilidades civiles o penales. El objetivo del atacante consiste en lograr acceso al sistema suplantando a un usuario, haciéndose con los privilegios del administrador o por otros métodos más sofisticados.

Se podría pensar que todo esto está únicamente al alcance de auténticos genios de la programación o de esa pequeña élite de *hackers* –menos de mil en todo el mundo– capacitados para detectar defectos de seguridad en un laberinto de líneas de código en ensamblador. En la práctica cualquiera puede hacerlo sin necesidad de tener grandes conocimientos de informática. Existen *suites* de software que incluyen todos los *exploits* o ataques habituales y que son relativamente fáciles de manejar. Paradójicamente estas herramientas no han sido creadas por delincuentes sino por *hackers* éticos con el propósito de comprobar sistemas y reparar huecos de seguridad en los mismos. Sin embargo los maleantes informáticos también se sirven de ellas para sus propios fines. Un agresor que

adquiera práctica en el manejo de Metasploit o Backtrack puede llegar a ser tan peligroso como los legendarios *hackers* de antaño.

### 1.4.5 Puertas traseras

Una vez que consiga acceder al interior de un sistema, el atacante podrá espiar el tráfico de la red, instalar software parásito o robar datos. Normalmente lo primero que hará es consolidar sus posiciones. Antes de que el administrador legítimo instale nuevos parches de seguridad o corrija los defectos de configuración que hicieron posible asaltar el sistema con éxito, el intruso creará nuevas cuentas con privilegios de administrador o lanzará algún proceso oculto que mantenga abierta la comunicación con el exterior. Si el objetivo, una vez tomado, parece bueno para convertirlo en una base de operaciones permanente, será el propio atacante quien instale los parches de seguridad que faltan para evitar que otros vengan detrás de él perturbando la paz del sistema y llamando la atención del administrador.

Un método habitual consiste en instalar un *rootkit* que haga posible un control remoto del ordenador para utilizarlo con diversos fines –robo de datos, pertenencia a una *botnet* o dedicar parte de su disco duro al alojamiento de *warez* o pornografía infantil– sin que las mismas figuren listadas por el comando ls de Linux o el administrador de procesos del sistema de Windows. Los *rootkits*, a diferencia de virus y troyanos, son malignos no por su potencial destructivo o perturbador sino por todo lo contrario, por su capacidad para pasar desapercibidos, y también por lo difícil que resulta eliminarlos. Esta complicación alcanza al extremo de tener que recurrir al formateo de discos duros y a la reinstalación del sistema operativo.

### 1.4.6 Borrando huellas

Si un atacante ha conseguido privilegios de administrador intentará suprimir de los *logs* o archivos de registro del sistema todos aquellos datos que pudieran ser reveladores de su actividad, o al menos los alterará para disimular la intrusión. Esto implica borrar *logs* e históricos de Internet, reiniciar sistemas de registro de eventos del sistema y otras manipulaciones similares.

## Capítulo 2

# LA INVESTIGACIÓN FORENSE

### 2.1 ETAPAS DE UNA INVESTIGACIÓN FORENSE

Una vez intervenido todo el material y tomadas las medidas necesarias para asegurar la cadena de custodia comienza el trabajo propiamente dicho del informático forense. Por lo general la investigación nunca ha de llevarse a cabo sobre los soportes de datos originales. Si se trata de un disco duro o un soporte que admite acceso en modo escritura, la razón para ello es obvia: cuando el investigador monta en su estación de trabajo las particiones de un soporte de datos sospechoso con el propósito de examinarlas, aquellas pueden experimentar algunos cambios. Esto proporcionaría al abogado de la parte contraria argumentos para impugnar la evidencia alegando que no se ha mantenido la cadena de custodia. Los modernos sistemas de archivos –NTFS (en entornos Windows), ext3, ext4, ReiserFS (para Linux), o HFS+ (Apple OSX)– disponen de una funcionalidad de *journaling* o verificación de transacciones para hacer posible la autorreparación de las estructuras de datos en caso de fallo o apagado irregular del sistema. Al ejecutar su tarea, el *journaling* modifica determinados archivos de registro. Esto sucede no solo tras un apagado irregular, sino cada vez que se monta la partición. Por similares razones jamás debe encenderse un ordenador que haya sido intervenido para someterlo a un análisis forense. Aun tratándose de un sistema sin *journaling*, el solo hecho de arrancarlo puede modificar varios centenares de archivos en la partición donde se encuentra instalado el sistema operativo.

Toda precaución es poca cuando se trabaja con elementos de evidencia digital. Aunque los archivos alterados no sean significativos y contengan únicamente información relativa al funcionamiento del sistema, resultará muy difícil defenderlo ante un tribunal en el que la otra parte expone la tesis opuesta: que la evidencia está alterada, existen indicios de ruptura en la cadena de custodia o hemos sido negligentes en el tratamiento de las pruebas.

Trabajar con imágenes forenses en vez de con el medio original también es una buena práctica en relación con aquellos soportes que por ser de solo lectura excluyan de antemano cualquier modificación del contenido, como por ejemplo CD, DVD o discos duros con la pestaña de seguridad activada. La destrucción accidental de un soporte de datos aportado como medio probatorio, aunque sea por causas que no tengan que ver con su procesamiento informático (calor, caída involuntaria o atasco dentro de la unidad de lectura) puede resultar desastrosa para el caso.

#### 2.1.1 Adquisición (Imaging)

En Informática Forense se denomina adquisición al procedimiento que permite obtener los medios digitales que han de ser sometidos posteriormente a análisis. Ya hemos dicho que por norma no se trabaja con el soporte original sino con una copia a bajo nivel del mismo. La copia puede estar realizada en un soporte físico duplicado o consistir en un archivo de ordenador. Dicho archivo no consiste en un simple *backup* sino en una imagen completa del medio de almacenamiento de datos, incluyendo el espacio no asignado por los sistemas de archivos, los archivos borrados e incluso datos que pudieran haber existido antes de que el soporte fuese formateado. La imagen completa de un soporte incluye todas las particiones, los espacios de disco duro sin utilizar entre las mismas, la tabla de particiones, el sector de arranque e incluso zonas reservadas como la HPA (*Host Protected Area*) y la DCO (*Data Configuration Overlay*), generalmente inaccesibles al sistema y utilizadas por el fabricante para incluir información especial o reducir la capacidad de almacenamiento de un dispositivo por razones de diversa índole, generalmente tecnológicas o de marketing.

Para obtener una imagen a bajo nivel es necesario conectar el soporte de datos a una estación de trabajo provista de herramientas que permitan el acceso en modo de solo lectura. Mucho mejor sería forzar el modo de solo lectura mediante dispositivos de hardware que impidan cualquier operación de escritura (*write blocker*). Las herramientas de software utilizadas para la adquisición forense (dd,

EnCase, FTK, Air, etc.) deben ser capaces de funcionar independientemente de las estructuras de datos del sistema de archivo. El procedimiento suele ser el mismo para todos los medios: discos duros (internos, portátiles o de estado sólido), llaves USB, disquetes, tarjetas de memoria, CD, etc.

De cada uno de los soportes de datos deberá realizarse una suma de verificación a través de funciones *hash* aplicando algoritmos estándar (MD5 o SHA) admitidos por la generalidad de especialistas en la materia. Las funciones *hash* se emplean en criptografía por sus propiedades de cifrado asimétrico. Su característica más notable consiste en que a partir de los datos digitales de entrada –correspondientes en este caso a la imagen del medio adquirido– generan un código hexadecimal que varía de modo perceptible cuando el archivo original experimenta cualquier cambio, aunque sea un solo *byte*. Una vez realizado el *hash* y a partir del código que aquel proporciona, resulta del todo imposible recuperar los datos originales; tanto da si se trata de un archivo de ordenador, una contraseña o el contenido de un disco duro. De este modo el *hash*, que puede calcularse al vuelo mientras la imagen del soporte de datos está siendo copiada, actúa como certificado digital para la validación de los elementos de evidencia adquiridos por el investigador forense.

## 2.1.2 Análisis

El análisis consiste en la identificación, el estudio y la interpretación de los elementos de evidencia existentes en el soporte de datos. En esta etapa el investigador lleva a cabo un examen detallado de los sistemas de archivos, intenta detectar archivos sospechosos y analiza el contenido de los mismos, realiza operaciones de búsqueda de caracteres, elabora estadísticas y ejecuta otras tareas de investigación. La última etapa del trabajo consiste en interpretar los resultados y preparar el informe.

## 2.1.3 Presentación

Una vez que el investigador ha terminado su análisis deberá preparar sus resultados para que los mismos puedan ser compartidos con las personas que se van a encargar de utilizarlos con fines prácticos o hacerlos valer ante los tribunales. En el informe definitivo deberá dejarse constancia documental precisa de todas las operaciones realizadas, los elementos de evidencia localizados y cualquier otro aspecto de interés. Adviértase que hasta el momento no se ha hablado de evidencia, sino de elementos de evidencia. La razón reside en el carácter acumulativo de la prueba. En el transcurso de su actividad el investigador no deberá dejarse llevar en

ningún momento por la suposición de que su tarea consiste en descubrir la pista definitiva que permita resolver el caso con un brillante golpe de efecto. Por el contrario, su labor consiste en encontrar indicios y señales que se irán agregando a un inventario de artefactos con el objeto de interpretarlos de un modo profesional y circunspecto dentro del contexto técnico en el que desempeña su labor.

La presentación, como pronto se verá, lejos de ser un trámite formal para cumplir el expediente, posee una importancia decisiva de cara a las posteriores actuaciones judiciales o administrativas. Posiblemente el informe vaya destinado al jefe de seguridad de la empresa. Tal vez vaya a parar a un despacho de abogados o al departamento de regulación de daños de una compañía de seguros. Quizás el propio investigador forense se vea obligado a comparecer ante un tribunal para rendir cuentas de su trabajo.

## 2.1.4 La línea de tiempo

La línea de tiempo, de la cual volveremos a hablar más adelante, es un concepto que ayuda al investigador a comprender la evolución de los hechos y las relaciones de causa y efecto existentes en los mismos. Ningún análisis forense serio se puede permitir dejarla de lado. Idealmente consta de una sucesión de momentos indicados por algún tipo de indicio (por ejemplo entradas en archivos de registro, creación de un archivo en fecha y hora determinadas) en los que tuvieron lugar intentos de intrusión en el sistema u operaciones no autorizadas de cualquier otro tipo: el sospechoso inició sesión, ejecutó un comando para abrir la conexión con una máquina remota, creó, borró o modificó un archivo, etc. La línea de tiempo se construye a partir de todos aquellos elementos de evidencia que contengan información temporal fiable: marcas de tiempo (MAC) de archivos, fechas y horas halladas en metadatos de archivos y en *logs* del sistema, historiales de Internet, etc.

La elaboración de una línea de tiempo es un proceso predominantemente manual, aunque existen algunas herramientas que facilitan la labor extrayendo datos temporales del sistema y poniéndolos a disposición del investigador tabuladas o por medio de listas. Además de situar los hechos en una perspectiva cronológica, la línea de tiempo añade información contextual. Por ejemplo, las marcas de tiempo MAC de un archivo nos dicen en qué momento fue creado o modificado, y también cuándo sucedieron otros hechos que pueden estar relacionados a la misma operación o incluso otros que no han sido corroborados aún pero resultan necesarios en función de las características técnicas del sistema. La línea de tiempo ayuda a excluir la ambigüedad en las afirmaciones y aumenta la confianza del investigador con respecto a su propio análisis de los datos disponibles.

## 2.2 REQUISITOS DE LA INVESTIGACIÓN FORENSE

De poco servirían las herramientas y los métodos utilizados por el investigador si no pudieran acreditar una cierta solvencia y validez ante el tribunal. Para ello es necesario que cumplan determinados requisitos. No olvide que los resultados de su labor investigadora deberán ser puestos a disposición de autoridades judiciales y otras instancias decisorias. Una herramienta forense, tanto si está homologada como si no, debe demostrar en su funcionamiento unos niveles de eficacia e integridad lo suficientemente altos para ganarse el respeto tanto de la comunidad profesional de investigadores como del personal que trabaja en la administración de justicia. Lo mismo cabe decir de los métodos de trabajo y la forma de exponer resultados. Alexander Geschonneck, analista del departamento de Informática Forense de la sociedad de auditores de contabilidad KPMG AG WPG, Berlín, y autor de un libro clásico sobre la materia (*Computerforensik*, dpunkt.verlag, Heidelberg, 2008), establece seis requerimientos esenciales que toda investigación debe cumplir sin falta.

- Aceptabilidad.
- Integridad.
- Credibilidad.
- Relación causa-efecto.
- Carácter repetible.
- Documentación.

### 2.2.1 Aceptabilidad

Las herramientas y métodos del investigador deberán ser conocidos y aceptados por los profesionales de su sector. La introducción de tecnologías innovadoras puede resultar problemática, porque no siempre lo más moderno es también lo mejor. Lo ideal sería que otros investigadores hubieran trabajado previamente con esos procedimientos y existan informes positivos sobre la eficacia de los mismos. Independientemente de la cualificación siempre se planteará una cuestión de confianza: si el método en cuestión es tan bueno y somos los únicos en servirnos de él, ¿por qué nadie más lo aplica?

### 2.2.2 Integridad

Las pruebas no deben sufrir alteraciones de ningún tipo. Generalmente el medio –disco duro, *pendrive* o CD/DVD– se precinta después de haber obtenido

tres copias cuyos *hashes* han de coincidir. Con una de ellas llevará a cabo su análisis el investigador. Otra se guardará como respaldo, y la tercera será puesta a disposición de la parte contraria para que esta pueda realizar sus propias averiguaciones, manifestar su posición al respecto o elaborar un contrainforme.

### 2.2.3 Credibilidad

Todo lo que se haga debe ser demostrable. No basta con utilizar un software del cual nada se sabe, salvo que si lo alimentamos con determinados datos siempre brotan de él determinados resultados. El investigador debe acreditar un conocimiento adecuado de sus herramientas para poder explicar de manera plausible lo que consigue de ellas.

### 2.2.4 Relación causa-efecto

Aunque no es cometido del investigador extraer conclusiones de ningún tipo sobre culpabilidad o responsabilidades de las personas que intervienen en los hechos, los métodos empleados por aquel deben hacer posible una explicación de los acontecimientos en términos de causa y efecto.

### 2.2.5 Carácter repetible

Este requisito se explica por sí mismo. Sean cuales fueren los métodos de trabajo empleados o la persona que realiza la investigación, los mismos datos de entrada deberán producir los mismos resultados.

### 2.2.6 Documentación

Cada paso dado por el investigador deberá disponer de una descripción detallada y exacta, al objeto de que los informes no puedan ser impugnados por culpa de ambigüedades o negligencias de ningún tipo. Especial cuidado deberá tenerse a la hora de documentar la cadena de custodia que es la parte más sensible de todo el proceso y la que con más facilidad podrá atacar la parte contraria en caso de localizar la menor irregularidad. Para ello sería conveniente que el investigador elaborase sus propias hojas de control de medios probatorios digitales, adjuntándolas debidamente al informe del caso. Estas hojas deben cumplimentarse para cada uno de los soportes de datos manipulados durante la investigación, y en ella habrán de constar claramente la denominación del objeto (disco duro, agenda digital, CD-ROM, listado de impresora, ordenador portátil, etc.), el número de unidades, el propietario (si es conocido). Además la hoja debe facilitar una descripción precisa del soporte y, en caso de que hubiera otras personas involucradas en el caso además del propio investigador, dejar constancia

documental precisa de quién, cuándo y por qué motivo concreto accede a los medios probatorios.

A modo de ejemplo se ofrece el siguiente modelo, que podrá ser modificado y complementado por el investigador en función de sus propias necesidades. Sea cual sea el sistema del cual se sirva, lo más importante a tener en cuenta es que la cadena de custodia debe quedar documentada de manera precisa, completa y sin huecos de ningún tipo.

<b>Hoja de control de medios probatorios</b>		Caso:		
Fecha:	Lugar de los hechos:	Nr.Id.:		
Horas:				
Investigador:	Testigo:			
Firma:	Firma:			
Objeto:	Número:	Descripción (tipo, fabricante, número de serie, características, etc.):		
<b>Control de accesos</b>				
<b>Objeto:</b>	<b>Fecha/Hora:</b>	<b>Entregado por:</b>	<b>Recibido por:</b>	<b>Motivo:</b>
		Nombre:	Nombre:	
		Organización:	Organización:	
		Firma:	Firma:	
		Nombre:	Nombre:	
		Organización:	Organización:	
		Firma:	Firma:	
		Nombre:	Nombre:	
		Organización:	Organización:	
		Firma:	Firma:	
		Nombre:	Nombre:	
		Organización:	Organización:	
		Firma:	Firma:	
<b>Protocolo de entrega final</b>				
<i>Operaciones (devolución al propietario, archivado, destrucción, etc.)</i>	<i>Destinatario de entrega, testigo(s):</i>			
	Nombre y apellido:	Firma:	Fecha:	
	1)			
	2)			
	3)			
	4)			

Tabla 2.1. modelo de hoja para el control de la cadena de custodia

## 2.3 VALORACIÓN JURÍDICA DE LA PRUEBA DIGITAL

### 2.3.1 Interés legal de la prueba

Para el técnico informático la obtención de elementos de evidencia electrónica no plantea mayor dificultad que la que pueda presentarse a la hora de hacer un *backup*: conectar discos, encender el ordenador, un par de clics con el ratón o teclear unas cuantas órdenes en línea de comando y sentarse a esperar hasta que el sistema termina. Pero el ámbito jurídico, al cual va destinada la evidencia, no se rige por la mentalidad lineal del ingeniero, sino por las alambicadas categorías del derecho. El destino de toda prueba digital es ser expuesta ante los tribunales, y su solvencia jurídica dependerá de las circunstancias en que haya sido obtenida.

### 2.3.2 Prueba física y prueba personal

Pruebas físicas –no en sentido literal, sino en el de la validez jurídica de las mismas– pueden ser discos duros, archivos de registro, informes periciales o huellas dactilares. Esta evidencia es llevada al proceso por personas que tienen que explicarlas y hacer valer su carácter probatorio en relación con los hechos juzgados. Por sí misma la prueba física, es decir, el objeto evidencial hallado en el lugar de los hechos, no demuestra nada. Haber encontrado huellas dactilares en el arma homicida no implica que quien las dejó cometiera el crimen necesariamente. También cabe pensar que tuvo el arma en sus manos en un momento anterior, para limpiarla o darle grasa. Por el contrario el verdadero agresor podría haber usado unos guantes para disparar. Del mismo modo hallar archivos de pornografía infantil en un ordenador no inculpa automáticamente a su propietario. Quizás sin saberlo él una red de ciberdelincuentes ha estado utilizando parte de su disco duro como almacén de materiales ilícitos a través de Internet.

La fuerza probatoria no se manifiesta hasta que intervienen las personas que hallaron la evidencia o aquellas que han de explicarla en relación con los hechos que se juzgan. Existe por lo tanto una estrecha relación entre la prueba física (disco duro, archivo de registro, mensaje de correo electrónico, volcado de memoria, fotografía digital) y la prueba personal (intervención del investigador forense ante el juez). El carácter probatorio de una evidencia, ya sea electrónica o de cualquier otro tipo, depende de la cualificación profesional y la solvencia de la persona encargada de explicarla en el proceso.

### 2.3.3 Cualificación del investigador forense

La presentación de los elementos de evidencia y los métodos utilizados en la consecución de aquellos van a ser comprobados por el tribunal minuciosamente. Pierde credibilidad quien aporta pruebas de forma inadecuada, realizando afirmaciones rebatibles o divagando caprichosamente acerca de la evidencia y sus posibles significados. El juez no tomará en serio al investigador que de manera imprudente y basándose en su propio convencimiento realice afirmaciones audaces relativas a la culpabilidad o inocencia del acusado. Decidirlo es competencia exclusiva de la administración de Justicia. Si el investigador no realiza bien su trabajo la prueba puede quedar invalidada en el proceso.

Solvencia y credibilidad de la persona constituyen elementos esenciales de la prueba. La persona que presenta y explica la prueba es tan importante como la prueba en sí. Un buen informe puede arruinarse por la defensa impropia del mismo, por ejemplo hablando de meras suposiciones como si fueran hechos irrefutables. Por el contrario una intervención austera y profesional, acompañada de una documentación minuciosa y alegaciones objetivas, resulta de gran ayuda a la hora de presentar pruebas electrónicas ante un tribunal.

### 2.3.4 La adquisición: fase crucial

Es en la obtención de los elementos de evidencia donde las buenas prácticas han de observarse con mayor rigor. El punto de partida de la investigación forense lo constituye la imagen a bajo nivel mediante flujo de bits obtenida a partir del soporte de datos. Conviene recordar que no estamos recuperando una tesis doctoral perdida por un profesor universitario a consecuencia de un fallo en el disco duro, ni las fotos de las vacaciones que se nos borraron accidentalmente por culpa de un virus. Estamos a punto de tomar parte en un proceso que puede tener graves consecuencias cuando lo que se juzga son responsabilidades penales. La pericia técnica no es suficiente. De lo que se trata es de preservar elementos de evidencia, asegurar una cadena de custodia y realizar duplicados exactos resistentes a las sumas de verificación y los alegatos de la parte contraria.

## Capítulo 3

# SOPORTES DE DATOS

En el transcurso de los últimos años la Informática Forense ha evolucionado con la incorporación de nuevas tecnologías y métodos de investigación –volcados de memoria RAM, adquisición de dispositivos móviles, análisis forense de redes y *cloud computing*, etc.–. No obstante existe un campo de actividad primordial al que se puede considerar “clásico”. La investigación de soportes de datos continúa siendo un elemento clave y una de las principales fuentes de evidencia. Esta situación se mantendrá en tanto no tengan lugar cambios revolucionarios en la tecnología (quizás mediante el traslado masivo a la nube o la aparición del ordenador cuántico).

Cuando hablamos de soportes de datos, el primer objeto que se nos viene a la mente es el disco duro del ordenador, con su apilamiento de platos movidos por un motor eléctrico, sus cabezales de lectura y su placa controladora atornillada en la parte baja. El concepto de soporte de datos se extiende a discos duros externos, iPods, CD-ROM y DVD, llaves USB, reproductores MP3, iPhones, smartphones Android, agendas electrónicas, teléfonos móviles, tarjetas de memoria y en general a todos aquellos dispositivos que sirvan para el almacenamiento permanente de archivos informáticos.

### 3.1 PROCEDIMIENTOS DE ADQUISICIÓN

Como ya se ha dicho el informático forense no trabaja con el soporte original, sino con una imagen a bajo nivel adquirida a partir del mismo. El soporte es precintado y guardado en el depósito de pruebas. La imagen se utiliza para llevar

a cabo las tareas de análisis, elaborar informes u obtener duplicados de la misma para la otra parte o por razones de seguridad.

Para adquirir un soporte es necesario establecer una conexión hardware. El método habitual consiste en desmontar el disco duro, llevarlo al laboratorio y conectarlo a una estación de trabajo forense. Si se trata de un disco IDE se puede conectar en el primer interfaz IDE como Slave (dando por supuesto que en el Master está el disco con el sistema operativo y las herramientas de investigación forense) o en el segundo interfaz IDE como Master o Slave, dependiendo de lo que tengamos instalado –grabadora DVD o un disco duro de gran capacidad para almacenar la imagen–. No olvide configurar los *jumpers* adecuadamente. Si el disco que queremos adquirir es de tipo SATA se podrá conectar a cualesquiera de los interfaces SATA que tengamos disponibles.

También se puede emplear un adaptador USB o Firewire para la lectura de discos IDE y/o SATA. Para discos duros SCSI necesitaríamos una tarjeta adaptadora y un cable de datos. La adquisición de discos duros externos, *pendrives*, iPods o cualesquiera otros dispositivos con interfaz USB se lleva a cabo a través de los conectores USB disponibles en el ordenador. Los CD y DVD se copiarán utilizando la grabadora. Las tarjetas de memoria deberán adquirirse a través de adaptadores conectados a un interfaz USB, etcétera. En todos los casos se tomarán precauciones para evitar que el sistema operativo o las aplicaciones accedan accidentalmente a los soportes adquiridos, modificando archivos o produciendo el más mínimo cambio en sus estructuras de datos. Por este motivo la adquisición se llevará a cabo a través de un bloqueador de escritura o *write blocker* (figura 3.1) o bien mediante un software especial que excluya el acceso en modo escritura.



Figura 3.1. Disco duro conectado a un bloqueador de escritura

El soporte de datos también puede ser adquirido sin necesidad de extraerlo del ordenador sospechoso. Para ello es necesario iniciar este último con un disco de arranque EnCase provisto de Linen (figura 3.2) o un Live CD Linux que incluya herramientas para la copia a bajo nivel. En tal caso deberán tomarse precauciones para evitar la escritura accidental en los soportes. Así mismo, en determinados casos en que la organización que ha sufrido el incidente no se pueda permitir mantener apagado el ordenador por tratarse de un servicio imprescindible, la adquisición se puede hacer con la máquina en funcionamiento a través de la red local mediante NetCat o CryptCat. Una modalidad reciente de adquisición es la que se suele llevar a cabo a través de red local en el contexto de las denominadas “investigaciones silenciosas”. Las últimas versiones del software de investigación EnCase, del que hablaremos más adelante, ofrecen la posibilidad de vigilar la actividad y las estructuras de datos de una máquina sospechosa en tiempo real.



Figura 3.2. Interfaz de LinEn

### 3.1.1 EnCase & Linen

EnCase Forensics, de Guidance Software, se ha convertido en un estándar de referencia en el campo de la Informática Forense. También ha establecido un formato específico para la realización de imágenes a bajo nivel (EnCase Evidence File Format). Al realizar el duplicado forense los soportes de datos son almacenados en archivos con un tamaño máximo de 2 GB extrayendo un *hash* MD5 de todo el caudal de bits. La imagen se puede hacer desde la interfaz de usuario de EnCase, después de conectar el medio a través del correspondiente interfaz IDE, SATA o USB, o bien desde el propio ordenador sospechoso tras

haberlo iniciado con un sistema externo (disquete o CD Live). Para esta operación se utiliza Linen, una herramienta desarrollada y distribuida libremente por Guidance Software, que permite volcar el contenido a un segundo disco duro IDE o SATA o a dispositivos de almacenamiento externos a través de USB o Firewire.

Linen, diseñado para entregar su salida en un formato forense de aceptación universal que cumple las especificaciones EWF (*Expert Witness Format*) de ASR Data, está incluido en algunas distribuciones de Linux especializadas en seguridad informática como Backtrack, Wifislax o la española AdQuiere. También se puede obtener un disco de arranque Linen desde la interfaz de usuario de EnCase.

La figura muestra una captura de pantalla de una terminal Linux. La barra superior dice "igandekoa:dd". El terminal muestra la ejecución del comando dd: "igandekoa@igandekoa-desktop:~\$ sudo dd if=/dev/sdb1 conv=sync,noerror bs=64K of=imagen\_usb.dd1". Se detallan los resultados: "60929+0 registros de entrada", "60929+0 registros de salida", "3993042944 bytes (4,0 GB) copiados, 245,016 s, 16,3 MB/s". La barra inferior dice "igandekoa:dd".

Figura 3.3. Adquisición de un soporte de datos con dd

### 3.1.2 dd

Aunque a primera vista no lo parezca, dd (figura 3.3) es una de las herramientas en línea de comando más poderosas y versátiles de toda la historia de las ciencias de la computación. Originaria del mundo Unix e incluida en todas las distribuciones Linux actuales, el funcionamiento de dd, explicado de modo simple, consiste en tomar partes de un archivo y copiarlas a otro. La potencia de una operación tan trivial se deriva del principio de abstracción característico de los sistemas Unix, donde todos los elementos están representados por un archivo: directorios, particiones, periféricos y dispositivos de almacenamiento de datos. Así mismo dd forma parte de Apple OSX, por tratarse de un sistema operativo basado en Unix. También existe una versión para Windows.

Con dd podemos obtener una copia en bruto (RAW) del soporte de datos, que después podrá ser sometida a operaciones de *hash* y analizada con cualquier software de investigación forense. La sintaxis básica de empleo, suponiendo que se esté trabajando desde un entorno Linux, es esta:

```
dd if=/dev/sda of=imagen.dd
```

Donde if (*input file*) es la fuente de origen de los datos –en este caso la primera unidad de disco IDE o SATA vista por un sistema operativo basado en Linux Ubuntu–, y of (*output file*) el archivo resultante de la copia. El ejemplo anterior sirve para llevar a cabo una adquisición completa del medio, incluyendo el MBR (*Master Boot Record* o sector de arranque maestro de un disco duro) con la tabla de particiones, los sectores de arranque de las particiones, las particiones con sus sistemas de archivos correspondientes, el espacio sin asignar y los archivos borrados. dd dispone de opciones para afinar el proceso e incrementar su funcionalidad. Es posible por ejemplo especificar durante la copia un tamaño de bloque determinado, ordenar a la herramienta que omita un determinado número de sectores antes de comenzar la copia, o la forma en que ha de responder ante errores de lectura de datos, etc.

Con dd se puede copiar prácticamente cualquier tipo de medio especificando en el origen de datos su archivo característico (por ejemplo /dev/sdc para una unidad DVD/CD-ROM, /dev/fd0 para la disquetera). Si no interesa copiar el soporte completo podemos seleccionar partes concretas del mismo. En el supuesto de que se quiera extraer únicamente la primera partición del disco duro, debe teclearse:

```
dd if=/dev/sda1 of=imagen.dd1
```

Y si solamente interesa el sector de arranque del disco duro:

```
dd if=/dev/sda of=mbr.dd bs=512 count=1
```

Para más información sobre dd y sus opciones el lector dispone de su página de manual en Linux: man dd.

Con dd también se puede llevar a cabo borrados seguros de datos. Esta función resulta de gran utilidad porque con carácter previo a la intervención del forense es preciso “higienizar” los soportes de destino, limpiándolos de aquellos datos previos procedentes de adquisiciones anteriores que pudieran comprometer el carácter válido de la evidencia. Para eliminar toda la información existente en un medio de almacenamiento no sirven de nada los métodos de borrado habituales. Tampoco es de ayuda formatear el disco duro. Para conseguir un medio de destino

totalmente vacío es preciso cubrir con ceros toda la superficie disponible para el almacenamiento de datos. El comando dd nos permite conseguir de modo rápido y sencillo ese objetivo:

```
dd if=/dev/zero of=/dev/sda
```

Las investigaciones forenses trabajan con grandes volúmenes de datos, por lo cual se habrá de tener en cuenta la disponibilidad de espacio suficiente en los soportes de destino. En caso necesario será preciso utilizar compresión.

Cuando se utiliza dd en modo de superusuario o *root*, con privilegios totales de acceso a todas las funciones del sistema operativo y a todos los dispositivos conectados al ordenador, hay que tener cuidado con lo que se hace y repasar a conciencia todo lo que se escribe en línea de comando antes de pulsar la tecla ENTER. Un error accidental en el manejo de dd puede traer consigo no solo la destrucción de elementos de evidencia sino también daños irreparables en el sistema operativo o los datos de su estación de trabajo forense. A no ser que se disponga de una plataforma especial desprovista de software y datos útiles y destinada únicamente a la realización de pruebas, se recomienda no hacer experimentos con dd antes de haber adquirido sólidas nociones de Linux.

### 3.1.3 dcfldd, dc3dd y ddrescue

Todos estos programas están basados en dd, cuyo código fuente, por ser software libre, está a disposición de todo aquel que quiera adaptarlo para incorporar características nuevas. De este modo ha surgido un número de variantes de dd con funcionalidades útiles para la investigación forense. Así, por ejemplo, dcfldd extrae y valida *hashes* de forma paralela, lleva un registro de actividad y divide la imagen en fragmentos de tamaño fijo para facilitar su almacenamiento y manipulación.

Por su parte con dc3dd, desarrollado por Jesse Kornblum, investigador del Centro de Criminología Digital del Departamento de Defensa de EE.UU., no es propiamente una variante de dd, sino un parche que posibilita la incorporación de características nuevas de dd con mayor rapidez que dcfldd. dc3dd posee las mismas funcionalidades que dcfldd e incluye otras propias del programa principal dd no existentes en dcfldd.

Sin embargo, la variante de dd que el investigador encontrará de mayor interés, por su utilidad en determinadas situaciones, es ddrescue. Si un soporte de datos posee sectores defectuosos, los podemos adquirir con dd especificando la opción conv=noerror,sync. Al hacer esto las partes ilegibles serán reemplazadas por ceros. Sin embargo supongamos que un disco duro recibido como prueba está

muy deteriorado y sospechamos que podría dejar de funcionar de un momento a otro. En el proceso de copia secuencial por defecto de dd, con intentos repetidos de lectura de los sectores defectuosos, suponen una pérdida de tiempo y aceleran la degradación mecánica del dispositivo. Lo que nos interesa es recuperar la mayor cantidad posible de información, y para ello ddrescue copia en primer lugar los sectores legibles. Solamente cuando estos han sido adquiridos en su totalidad intenta leer las partes defectuosas extrayendo lo que pueda de ellas. Este procedimiento incrementa las posibilidades de extraer evidencia utilizable de un disco duro deteriorado.

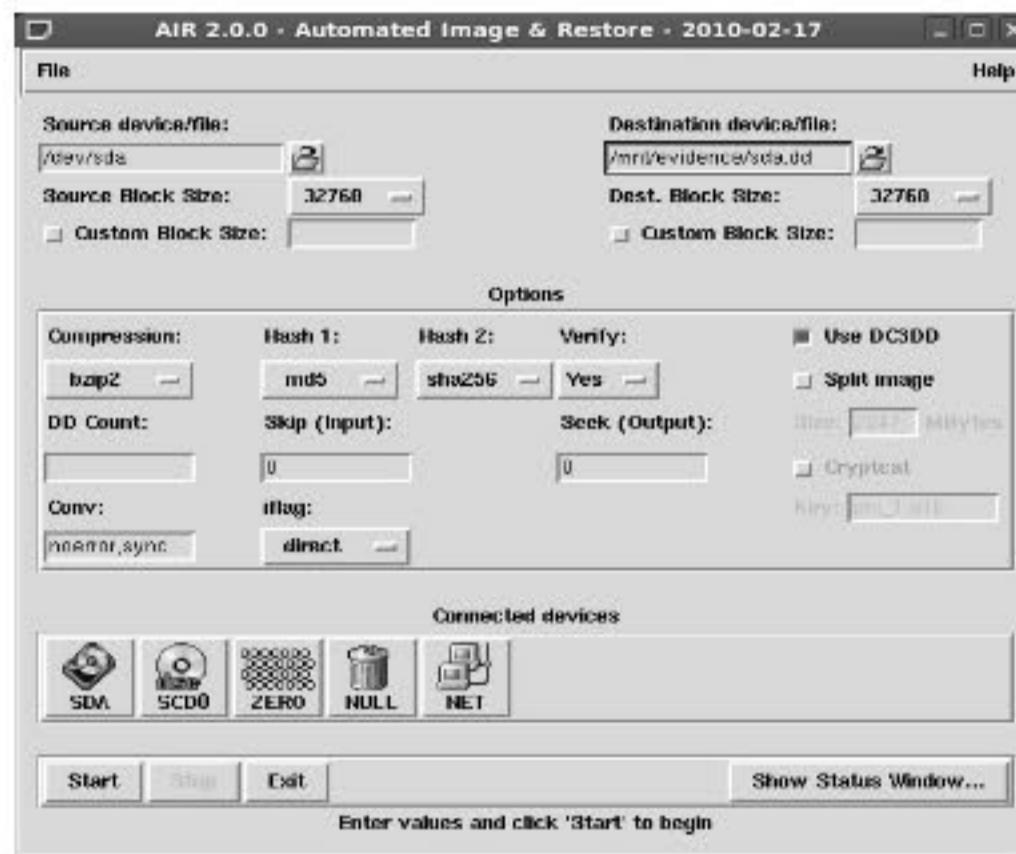


Figura 3.4. Interfaz de AIR

### 3.1.4 AIR

AIR (*Automated Image and Restore*) es un *front end* para dd y dc3dd, es decir, una interfaz gráfica que permite manejar de manera cómoda y sencilla, con las opciones más habituales, las utilidades en línea de comando mencionadas en el apartado anterior. Esta herramienta, además de elegir entre dd y dc3dd a la hora de realizar la imagen (figura 3.4), verifica la copia a través de *hashes* MD5 o SHA1/256/384/512, comprime y descomprime de la imagen mediante gzip/bzip2, ofrece la posibilidad de transferirla a un *host* remoto a través de redes TCP/IP, hace posible el borrado seguro de medios de destino y divide la imagen en archivos de un tamaño especificado para facilitar su manejo. AIR genera un historial cronológico y soporta sistemas de cintas con interfaz SCSI para copias de respaldo.

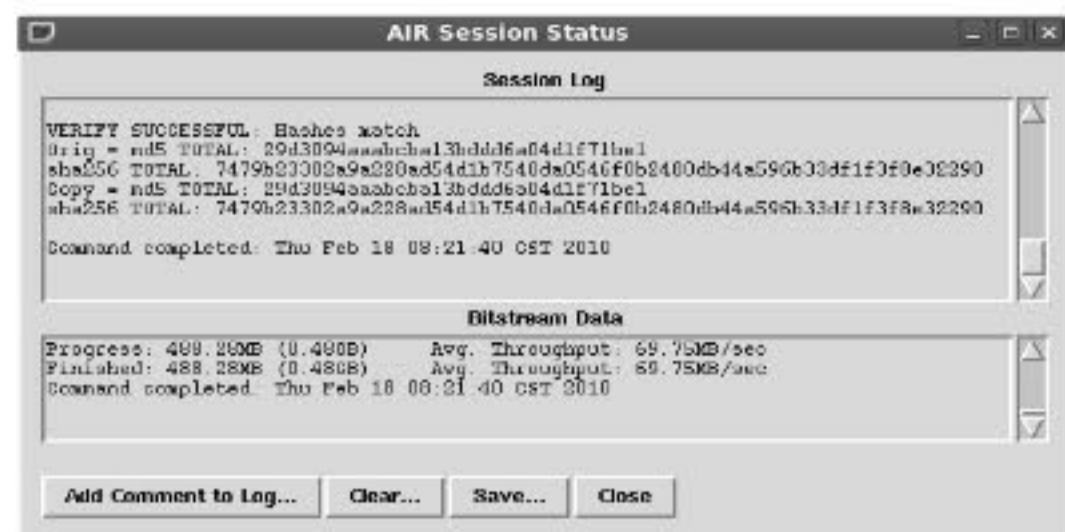


Figura 3.5. Informe de estado de AIR

### 3.1.5 Adquisición por hardware

Existen dispositivos dedicados exclusivamente a la copia de soportes de datos, principalmente discos duros con interfaces habituales IDE/SATA, que evitan tener que utilizar un ordenador en el proceso de adquisición forense. Estos aparatos alcanzan grandes velocidades de transferencia, extraen *hashes*, elaboran registros de actividad y admiten compresión. En la mayor parte de los casos se puede elegir como *output* bien un duplicado exacto del medio de origen, bien una colección de archivos en formatos estándar EnCase o FTK. Las ventajas de estos aparatos saltan a la vista y compensan de sobra un coste de adquisición que, gracias a la competencia entre los fabricantes de este tipo de hardware, es ya incluso inferior al de un PC de gama baja (figura 3.6).



Figura 3.6. Hardware para adquisición de soportes de datos

### 3.1.6 MD5 y SHA

MD5 y SHA son algoritmos de cifrado que a partir de un mensaje de tamaño variable, compuesto por un número determinado de caracteres alfanuméricos, de código binario o cualquier otro tipo, calculan una secuencia de caracteres de longitud fija. La cantidad máxima de información admitida como *input* (hasta 2 elevado a 64 bits para MD5) permite utilizar el contenido de discos duros de gran capacidad como entrada del algoritmo. En otras palabras, todo el contenido del medio –documentos, ejecutables, imágenes, estructuras del sistema de archivos– es procesado del mismo modo que un archivo de texto. El resultado es un número hexadecimal de 32 dígitos que cumple las propiedades del algoritmo de cifrado: (a) asimetría o imposibilidad de reconstruir el mensaje original a partir del *hash*; (b) es diferente para cada soporte de datos codificado, y (c) experimenta cambios drásticos con cualquier modificación –aunque sea un solo bit– que se haga en los datos de entrada. Aunque durante los últimos años se han descubierto algunos ataques contra estos algoritmos de cifrado, MD5 y SHA disfrutan de un amplio reconocimiento por parte de la comunidad de especialistas y forman parte de los procedimientos estándar utilizados por investigadores y agencias de seguridad de todo el mundo.

En la práctica forense el *hash* tiene una importancia crucial. En un juicio el abogado de la parte contraria hará lo posible por cuestionar la evidencia. Una de las estrategias más habituales consiste en sembrar dudas acerca de la integridad de los datos sobre los cuales se lleva a cabo la investigación. Si los *hashes* de la imagen y el soporte original no coinciden, entonces podrá argumentar que no existen suficientes garantías de que los datos analizados por el investigador sean copia exacta de la fuente. Quizás la adquisición del soporte no se ha llevado a cabo con las suficientes garantías de profesionalidad, la información está corrupta, o tal vez una mano negra ha introducido en ella modificaciones de forma interesada. De un modo u otro queda cuestionada la validez de las pruebas. Del mismo modo que toda investigación eficaz precisa mantener una cadena de custodia, la integridad de los datos constituye un elemento irrenunciable del método forense, al que deberá dedicarse una atención preferencial en el momento de adquirir los soportes de datos.

El procedimiento habitual consiste en hacer primeramente un *hash* del soporte original. Acto seguido se obtiene una copia a bajo nivel y para terminar volvemos a extraer el *hash* de la copia. Los dos *hashes* deberán ser iguales. Toda coincidencia posterior con estos *hashes*, en el transcurso de operaciones de análisis y duplicación de datos, equivaldrá a una constatación de que las pruebas que se están manejando durante las diligencias judiciales son idénticas al contenido del soporte original. En la práctica los *hashes* se extraen y comparan automáticamente

durante el mismo proceso de adquisición a través de herramientas especializadas como AIR.

### 3.1.7 Cálculo de MD5 con Linux

He aquí un ejemplo de cómo obtener y verificar el *hash* de un soporte de datos en entornos Linux:

```
md5sum /dev/sda
```

(o bien /dev/sdb, /dev/sdd, etc., dependiendo de cuál sea el interfaz en el que hayamos conectado el disco duro). Acto seguido se lleva a cabo la adquisición del soporte:

```
dd if=/dev/sda of=imagen.dd
```

Y finalmente se extrae el *hash* de la imagen:

```
md5sum imagen.dd
```

Los dos *hashes* deben coincidir. Por cierto, a la hora de obtener los *hashes* del soporte original y la imagen para compararlos no olvide que debe ser congruente en la denominación de los parámetros. Si ha realizado una copia a bajo nivel de una partición (/dev/sda1) y después calcula el *hash* de todo el medio (/dev/sda), lo sorprendente sería que los *hashes* coincidieran, por más que el trabajo de adquisición forense y mantenimiento de la cadena de custodia haya sido impecable.

## 3.2 DISCOS, PARTICIONES Y SISTEMAS DE ARCHIVOS

Una vez adquirido el soporte de datos y realizadas las copias de seguridad el investigador pasará a la segunda fase: el análisis de datos. Para ello, además de disponer de herramientas software cualificadas, deberá estar familiarizado con las diferentes tecnologías de almacenamiento: tipos de discos duros, interfaces de conexión, montajes RAID, volúmenes y sistemas de archivos.

Para entender bien qué es un sistema de archivos el lector debe plantearse la pregunta siguiente: habiendo tantos miles de archivos en su disco duro, ¿cómo se las arregla el sistema operativo para encontrarlos? ¿Cómo hace para seguir la pista de cada programa ejecutable, de cada biblioteca o de cada documento creado por el usuario? ¿Cómo consigue localizar sus fragmentos, ensamblarlos y presentarlos al usuario de manera casi instantánea y sin equivocarse una sola vez?

Lógicamente tiene que haber estructuras que contengan los nombres de los archivos acompañados de datos característicos como tiempos de creación y modificación de los mismos: un registro por cada archivo, con un número de campos determinado y todos organizados de manera coherente al igual que en una base de datos. Cada uno de estos registros deberá incluir una referencia exacta a los sectores del disco en los que se guarda el contenido del archivo –código ejecutable, texto, imágenes o sonido–, o al menos información que haga posible acceder a dicho archivo a través de tablas especiales. Todas estas estructuras deberán estar gestionadas y actualizadas por un controlador especial al que solo el sistema operativo tenga acceso. Y también tendría que haber otras estructuras de datos dedicadas a funciones complementarias, como por ejemplo llevar un registro de sectores y bloques disponibles, información de respaldo que permita reconstruir el sistema de archivos en caso de fallo y archivos especiales con marcas de verificación para completar operaciones incompletas tras un apagado brusco o irregular del sistema.

### 3.2.1 NTFS

NTFS fue desarrollado por Microsoft para los primeros sistemas operativos de la serie NT. Actualmente se utiliza en ordenadores equipados con Windows XP, Vista o 7. El 90% de los ordenadores de sobremesa funciona con alguna de las versiones del sistema operativo de Microsoft. Si citamos a NTFS en primer lugar ello se debe a que el investigador forense, a no ser que esté especializado en el análisis de plataformas específicas, se verá obligado a trabajar con este sistema con mayor frecuencia que con ningún otro. La probabilidad de que su primer caso tenga que ver con ordenadores provistos de un sistema de archivos NTFS es correspondientemente alta.

NTFS, que actualmente se encuentra en su versión 3.1, dispone de características avanzadas de seguridad como permisos de archivos, encriptación, cuotas de disco y registro de transacciones (*journaling*), que evitan la corrupción de datos reduciendo el esfuerzo de mantenimiento del sistema de archivos. Su elemento clave o estructura de datos principal es una tabla maestra de archivos (Mft) con entradas de longitud fija, cada una de las cuales contiene todos los datos necesarios para gestionar el archivo correspondiente: nombre y tipo de archivo, permisos, marcas de tiempo MAC (creación, acceso y modificación), localización de los bloques (*data runs*) donde se encuentran grabados los datos que forman parte del contenido del archivo e incluso flujos alternativos de datos (ADS: *Alternate Data Streams*), una característica especial de funcionamiento que permite asociar a un archivo bloques de datos adicionales además de su contenido.

Además de la Mft, NTFS dispone de un número de archivos auxiliares, entre ellos una copia de respaldo de las primeras cuatro entradas de la Mft, cuyo propósito consiste en hacer posible la reconstrucción del sistema en caso de pérdida accidental o intencionada de la tabla de particiones. También dispone de un mapa de bits para indicar qué bloques de datos están ocupados y cuáles se encuentran disponibles para alojar nuevos archivos, un archivo de información referente a cuotas de disco asignadas, tablas con descriptores de seguridad, etc. En el capítulo posterior sobre análisis forense de sistemas MS-Windows veremos la forma práctica de trabajar con particiones NTFS en una investigación.

### 3.2.2 FAT

FAT es el primer sistema de archivos de uso generalizado. Comenzó a utilizarse en los primitivos ordenadores de sobremesa equipados con el sistema operativo MSDOS. Su historial de servicios continuó bajo Windows 3.1, las primeras versiones de Windows basadas en interfaces gráficos, 95, 98, Millennium e incluso XP –que admite la opción entre FAT y NTFS–. Los sistemas de servidor (Windows 2003 y 2008) y de usuario actuales (Vista y 7) funcionan exclusivamente con NTFS (al menos en la partición del sistema), porque de otro modo no resultaría posible aprovechar las características de seguridad avanzadas de Windows.

FAT localiza los archivos a través de entradas de directorio que hacen referencia tanto a directorios como a archivos. Estas entradas incluyen el nombre del archivo o del directorio –junto con extensiones adicionales para nombres de 8 caracteres + 3 correspondientes a la extensión–, marcas de tiempo, tamaño del archivo, bits de características especiales que indican si el archivo es de sistema, solo lectura u oculto. Cada entrada de directorio indica la posición del primer bloque de datos o *cluster* del archivo e incluye un descriptor que apunta a una posición determinada de una tabla denominada FAT (*File Allocation Table* o Tabla de Asignación de Archivos).

La FAT mapea el espacio asignado a los archivos de tal manera que mediante un encadenamiento de entradas es capaz de marcar la localización exacta de cada bloque del archivo en el disco duro. La posición del primer bloque de datos o *cluster* se halla indicada en la entrada de directorio correspondiente al archivo, que remite a una posición de la FAT en la que se encuentra la referencia al segundo *cluster*, y así sucesivamente hasta incluir los bloques de datos correspondientes al contenido completo del archivo. El último bloque de datos o *cluster* se indica mediante el carácter hexadecimal FFFF, que es como una especie de furgón de cola que señala el final de archivo. En otro lugar de la partición, y por motivos de seguridad, se guarda un duplicado de la FAT, que se actualiza de manera

sincronizada para que en caso de pérdida o corrupción de cualesquiera de las dos FAT el sistema operativo pueda reconstruir la tabla de asignación de archivos dañada con la información redundante disponible en la otra.

Existen tres versiones de FAT, que difieren en la longitud de las entradas de la tabla de asignación de archivos. FAT12 se empleaba en los antiguos discos flexibles. FAT16 es típica de los primeros PC con MSDOS, con 8 caracteres para los nombres de archivo y 3 para la extensión. FAT32, un perfeccionamiento de la anterior, supuso en su tiempo un claro avance en el entorno de Windows 9x, permitiendo asignar a los archivos nombres largos y crear particiones mayores de 2 GB (que era el límite de tamaño con FAT16). Es conveniente que el investigador esté al tanto de estos detalles e incluso se documente en profundidad sobre las principales diferencias entre los tres tipos de partición FAT, porque los abogados de la parte contraria son muy aficionados a servirse de ellos para crear confusión en la defensa de informes forenses.

Comparado con otros sistemas de archivos más modernos como NTFS, HFS+ y la serie ext3/4 del entorno Linux, FAT presenta algunas limitaciones. Además de no disponer de características de seguridad ni de *journaling*, es poco flexible y no aprovecha de manera óptima el espacio en disco, debido a la desmesurada longitud de los *clusters* en particiones de gran tamaño, que hace que un archivo de texto de unos pocos caracteres ocupe el mismo espacio físico en disco que un documento grande de 60 KBytes. Otra de las desventajas de FAT consiste en favorecer la fragmentación de archivos, lo cual penaliza el rendimiento del sistema y hace necesario el uso de herramientas de software para desfragmentar el disco duro de vez en cuando.

Pese a ello continúa siendo ampliamente utilizado en todo tipo de dispositivos y soportes destinados a la informática de consumo: llaves USB, tarjetas de memoria para cámaras fotográficas, reproductores MP3, etc. En este sector la sencillez de FAT supone una ventaja con respecto a sistemas más avanzados, al permitir un mayor rendimiento en las operaciones de lectura y escritura, así como una movilidad sin límite de unos dispositivos a otros sin problemas ni bloqueos causados por características de seguridad o permisos de archivos.

El lector mismo lo podrá comprobar si en vez de utilizar su llave USB de 8 GB preformateada con FAT 32, es decir, tal y como viene de fábrica, crea en ella una partición NTFS e intenta trasladar archivos de unos ordenadores a otros. Con los sistemas de archivos ext2/3/4 característicos de Linux los resultados son aún más frustrantes. En cambio FAT, al carecer de permisos y seguridad, jamás dará problemas en este sentido.

No habiendo perspectivas de que se imponga un sistema más práctico, es de esperar que la hegemonía de FAT aún persista durante varios años.

### 3.2.3 ext2, ext3, ext4

Del mismo modo que Windows se apoya en NTFS y FAT, Unix/Linux utiliza también sus propios sistemas de archivos. Durante muchos años ext2 fue estándar en la mayor parte de las distribuciones Linux, pero recientemente se ha impuesto ext3, el cual a su vez dejará paso en breve al nuevo sistema ext4, ambos con características de *journaling*. Estos sistemas de archivos se basan en conceptos del mundo Unix. El disco duro se divide en particiones y estas a su vez en grupos, que vienen a ser una especie de particiones de particiones. El objetivo de esta compartimentación, en lugar de gestionar un espacio continuo, consiste en reducir la fragmentación de archivos. En todo sistema de archivos ext2 hay un superbloque que contiene los metadatos relativos al sistema. Cada uno de los grupos en los que se encuentra dividida la partición dispone a su vez de su propio superbloque, un descriptor de grupos, un mapa de bits para indicar la disponibilidad de bloques de datos dentro del grupo, otro mapa de bits para los *inodes*, una tabla de *inodes* y finalmente los bloques de datos para alojar en ellos el contenido de los archivos: documentos, código ejecutable, multimedia, etc.

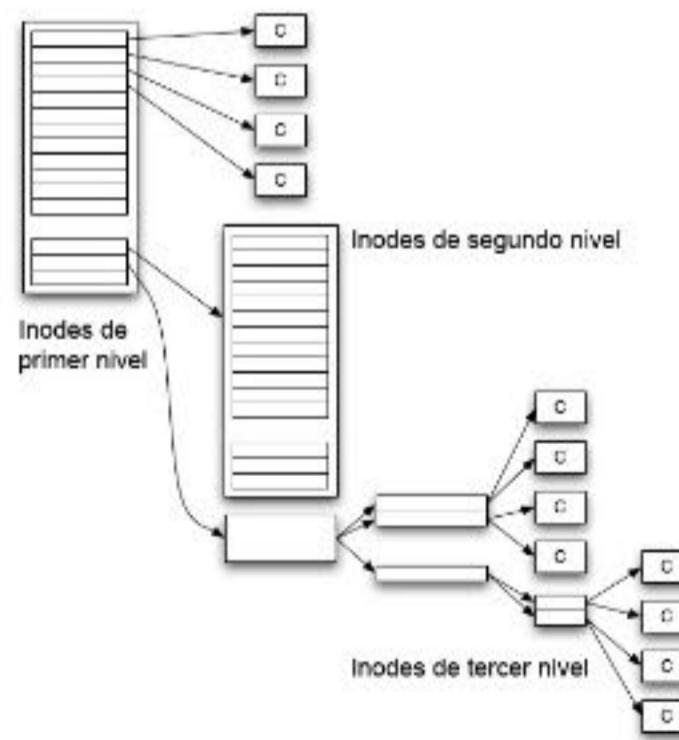


Figura 3.7. Sistema de localización de bloques mediante inodes

Los archivos están representados por estructuras de datos denominadas *inodes* (figura 3.7), los cuales además de describir las características del archivo —marcas temporales, permisos de acceso, etc.— apuntan a los bloques de datos. Cada *inode* dispone de 15 punteros de señalización a bloques de datos, con 13 de

ellos disponibles de modo efectivo para dicha función. No es número suficiente para abarcar los bloques de datos correspondientes a un archivo de tamaño razonable. Por consiguiente, si el archivo tuviera una extensión superior a los 13 bloques de datos, el puntero número 14 se utilizará para señalar a un “bloque indirecto” que dispone de otros 13 punteros para señalar a bloques de datos. Y si aún no bastara, el puntero número 15 referenciará a otro “bloque indirecto doble”, con punteros a otros bloques, cada uno de los cuales incluye a su vez otros 15 punteros. Un sistema de archivos de estas características presentará una constelación de *inodes* desperdigados por toda la partición. Para unificarlos en una estructura lógica es por lo que se creó la tabla de *inodes*.

Pese a su probada eficacia en la gestión de los datos un sistema basado únicamente en *inodes* resulta de poca utilidad para el usuario. Imagine que para llegar a un documento determinado, por ejemplo su declaración de la renta o el acta de la última reunión de la comunidad, hubiera de tener apuntado en alguna parte su *inode* correspondiente: 19134. Resultaría muy poco práctico. Por consiguiente, al igual que en NTFS (mediante las entradas de la Mft) o en FAT (con las entradas de directorio), se precisa un modo de relacionar cada *inode* con una denominación autoexplicativa. Aquí es donde intervienen los directorios. Estos no son sino archivos cuyo contenido consiste en una lista de entradas y punteros que señalan a los archivos pertenecientes a ese directorio. Un directorio de vital importancia en los sistemas Linux es “/”, o directorio raíz, que ocupa el lugar más alto de la jerarquía, y del cual penden todos los demás niveles. Este directorio raíz se encuentra siempre asociado al mismo *inode*, de modo que a partir de él resulta posible reconstruir la jerarquía completa de directorios y archivos durante el arranque del sistema operativo.

A su vez los subdirectorios no son otra cosa que enlaces a otros archivos con entradas de directorio para los archivos respectivos. Y así sucesivamente. En cada directorio existen dos entradas especiales, representadas por un punto (.) y dos puntos (..). Son los punteros de identidad –este directorio– y precedencia –directorío de nivel inmediatamente superior-. Estas entradas se generan en el momento de crearse el subdirectorio y no pueden ser eliminadas.

### 3.2.4 HFS, HFS+, JFS, ReiserFS, etc.

Los sistemas de archivos que acabamos de ver son los que más se utilizan en la actualidad, pero no los únicos ni mucho menos. En las cuatro décadas transcurridas desde los comienzos de la Revolución Informática han existido numerosos sistemas de archivos, con los cuales el investigador podría encontrarse en algún momento de su vida profesional dependiendo de las circunstancias. Aquí va, a modo de ejemplo y sin que la relación siguiente pretenda ser completa, una

sopa de acrónimos: AOFS, AAFS y ASFS (para los extintos ordenadores Amiga), UMSDOS (sistema que permitía montar sistemas Linux en particiones FAT), Fossil (exclusivo del sistema operativo Plan 9 de los Laboratorios Bell), ISO 9660 (CD-ROM en solo lectura), UDF (DVD y algunos CD), ext y MINIX (para sistemas Unix antiguos), ZFS (Sun Microsystems), XFS y JFS (sistemas de alto rendimiento con *journaling*), ReiserFS y Reiser4 (sistemas con *journaling* para Linux).

La creciente popularidad de los dispositivos Apple (ordenadores de sobremesa Mac, *smartphones* y reproductores de medios iPod e iTouch) obliga al investigador a familiarizarse con una nueva especialidad de la Informática Forense dedicada al análisis de soportes de datos utilizados con el hardware y los sistemas operativos de la marca Apple. Gracias a su potencia, versatilidad y capacidad para trabajar con diferentes sistemas de archivos, los ordenadores con el logotipo de la manzana también constituyen no solo objetos de investigación cada vez más habituales, sino también excelentes plataformas para la investigación forense. No está de más que el investigador adquiera práctica con los principales sistemas de archivos del entorno Apple: HFS para antiguos ordenadores McIntosh y HFS+ para los nuevos Apple OSX de sobremesa y portátiles. Este último también es el sistema de archivos por defecto para los dispositivos móviles de la casa Apple (iPhone, iPad e iTouch).

## 3.3 MODELO DE CAPAS

A la hora de estudiar un soporte de datos –disco duro, CD-ROM, llave USB, memoria *flash*, etc.– podemos seguir una metodología basada en niveles de funcionamiento similar al modelo de capas OSI de las redes informáticas. El nivel más bajo correspondería al hardware y a la tecnología de señales eléctricas, mientras que el más alto sería representativo de nociones conceptuales y abstractas más próximas al lenguaje humano –nombres de archivos-. Este enfoque, establecido por Brian Carrier en una obra de referencia ya clásica sobre investigación forense de sistemas de archivos, ofrece considerables ventajas conceptuales y puede ser aplicado a cualquier sistema de archivo y tecnología de soporte de datos.

### 3.3.1 Nivel 1: dispositivos físicos

En este primer y más bajo nivel, similar a la capa hardware de la pila de protocolos OSI, estaría comprendida la tecnología básica del soporte de datos: por ejemplo, tratándose de discos duros, aquí nos encontraríamos con elementos electromecánicos, circuitería de control y carcassas precintadas. Analizar discos duros no está al alcance de todo el mundo: se requiere disponer de salas limpias,

instrumental sofisticado, personal capacitado y un vasto *stock* de piezas de repuesto para las operaciones de rescate de datos cuando el soporte está dañado. La manipulación la suelen llevar a cabo empresas especializadas que cuentan con los necesarios equipamientos y una tecnología adecuada.

Menos problemático para el investigador es el tener que tratar con discos duros de estado sólido y soportes basados en memoria *flash* (llaves USB, tarjetas para cámara digital y chips de teléfonos móviles y *smartphones*), donde cualquier experto en microelectrónica puede solventar tareas de cierta complejidad con un instrumental asequible. Pero por lo general, todo lo perteneciente al nivel de los dispositivos hardware es algo que viene dado de fábrica y sobre lo que el investigador no posee más capacidad de actuación que la relativa a la consulta de especificaciones en Internet o la contratación de un servicio especializado, generalmente a precios muy altos.

### 3.3.2 Nivel 2: volúmenes y particiones

El esquema de particionado de los discos duros es algo que genera interpretaciones confusas. Términos como “partición” y “volumen” suelen utilizarse indistintamente y de forma en ocasiones contradictoria. Indudablemente, y de modo similar a lo que sucedía en la famosa polémica en torno al año 2000 y los comienzos del nuevo milenio, la discusión tiene que ver más con la semántica que con la tecnología. Por lo general se admite que la diferencia entre particiones y volúmenes es la siguiente: cuando se trata de un disco duro estándar dividido en sectores (bloques de datos básicos), pistas (grupos de sectores que hacen un giro completo en el plato) y cilindros (el conjunto de pistas situadas en la misma posición de todos los platos), una partición consiste en un grupo de cilindros contiguos; por el contrario el volumen no tiene por qué ser contiguo, pudiendo estar compuesto por varias particiones situadas en lugares distintos del disco duro o incluso en discos duros distintos.

En el entorno del PC, por limitaciones no tanto de tecnología como de diseño de los sistemas operativos, los soportes de almacenamiento –discos duros, discos duros externos, tarjetas de memoria y llaves USB– admiten un máximo de cuatro particiones primarias. Si quisieramos tener más de cuatro (algo habitual en Linux y en sistemas con arranque dual Windows/Linux), una de las particiones primarias habrá de ser convertida en una partición extendida, para fraccionarla después en tantos volúmenes lógicos como se necesiten.

Identificar las particiones constituye un requisito fundamental en todo trabajo de investigación forense, sobre todo si el disco está deteriorado o el sospechoso ha borrado el sector de arranque. Conocer bien el nivel correspondiente

a las particiones resulta de gran utilidad a la hora de adquirir un soporte. El investigador debe tener claro si la imagen a bajo nivel que acaba de copiar corresponde al medio completo o solo a una parte, ya que de ello depende la eficacia de la investigación o la posibilidad de pasar por alto elementos de evidencia decisivos.

A menudo la asignación de letras a particiones propia de Windows resulta algo confuso, puesto que el sistema operativo no muestra las particiones Linux ni las zonas de intercambio (*swap*). Para saber lo que contiene un disco sospechoso no basta con conectarlo a través de un bloqueador de escritura a una estación de trabajo Windows, sino que además es necesario disponer de herramientas especiales –típicamente un gestor de particiones o software de análisis forense como EnCase o FTK–. Si utilizamos Linux lo tenemos más fácil ya que podemos emplear el programa fdisk de Linux o las utilidades Sleuth Kit (de las cuales se hablará más adelante). Sin embargo, ni siquiera de ese modo existen garantías de llevar a cabo una adquisición completa. Los discos duros suelen disponer de zonas especiales como la HPA y el DCO, que los fabricantes incluyen para almacenar información sobre el dispositivo o reducir la capacidad de almacenamiento con el objeto de segmentar el mercado (en vez de producir medios de almacenamiento de datos con capacidades distintas resulta más barato fabricar todos los discos de la misma capacidad, por ejemplo 1 TB, y después reducirla mediante un DCO a tamaños de 750, 500, 320 GB, etc.).

### 3.3.3 Nivel 3: sistemas de archivos

En apartados anteriores se han descrito algunos de los sistemas de archivos utilizados con mayor frecuencia en la actualidad. La tarea del investigador consiste en identificarlos y examinar sus características. Esto es necesario no solo para acceder a los archivos del soporte, sino también para localizar elementos de evidencia. Un sistema de archivos organiza el contenido de su partición a través de metadatos y estructuras de control. El sistema de archivos se puede determinar de diversas formas. Podemos utilizar un gestor de particiones, el comando fdisk de Linux, herramientas en línea de comando o software de investigación forense. La búsqueda de evidencia dentro de una partición o un volumen deberá llevarse a cabo de acuerdo con las características del sistema de archivos para el que dicha partición ha sido formateada.

### 3.3.4 Nivel 4: bloques de datos

El bloque o *cluster* –como se le suele denominar en entornos Windows– es la unidad de menor tamaño disponible para el almacenamiento de datos en una partición que haya sido formateada con un sistema de archivos determinado. En el

nivel lógico más elemental los discos duros aparecen divididos en sectores de 512 *bytes*. Este esquema lo siguen por conveniencia otros tipos de soporte de datos que por sus características técnicas no tendrían necesariamente por qué hacerlo, ya que no disponen de platos giratorios divididos en cilindros y sectores ni cabezales de lectura, sino tan solo de chips de memoria *flash* o sistemas similares desprovistos de piezas móviles. La razón de que aun así lo hagan está en que de esa manera pueden utilizar la misma tecnología básica y los mismos *drivers* que se emplean para acceder a los sistemas de archivos implementados en los discos duros. Tanto en Unix como en Windows el bloque de datos característico se forma agregando sectores en potencias de 2, pudiendo tener una longitud de 1.024, 2.048, 4.096 *bytes*, etc., dependiendo de las características del sistema operativo, del tamaño total de la partición y a veces de las preferencias del usuario.

Los archivos, independientemente de la longitud en *bytes* de los mismos, se almacenan en disco ocupando bloques completos. Muy pronto veremos la importancia que tiene eso. En este nivel del modelo de capas es donde el investigador encontrará lo que más le interesa: los datos que integran el contenido del archivo. Si el bloque está asignado a un archivo gráfico, contendrá información codificada perteneciente a una imagen JPEG, PNG o de otros tipos. Si se trata de un documento hallaremos caracteres de texto y marcas de formato XML o Word. Y si corresponde a un ejecutable, el cargamento estará compuesto por instrucciones en VisualBasic, caracteres hexadecimales correspondientes a código binario o líneas de comentario del programa.

### 3.3.5 Nivel 5: metadatos

Los metadatos se definen por lo general como datos que hacen referencia a otros datos. No hay que confundir los metadatos del sistema de archivos con los de documentos PDF o MS-Office. En este último caso van incrustados en el documento, mientras que los metadatos de un sistema de archivos permanecen disociados de los archivos y forman parte exclusivamente de las estructuras de control del sistema de archivos. En Unix/Linux los metadatos se encuentran incluidos en los *inodes* mientras que Windows los guarda en las entradas de directorio (FAT) o en la Mft y sus archivos auxiliares (NTFS). El contenido y la estructuración de los metadatos dependen del sistema de archivos. Por lo general este nivel ofrece al investigador elementos de evidencia como marcas de tiempo (creación, acceso, modificación), identidad del propietario del archivo y punteros a los bloques de datos en los que se encuentra almacenado el contenido. En capítulos posteriores se discutirán características y elementos de evidencia de cada sistema de archivos.

### 3.3.6 Nivel 6: nombre de archivo

En la parte superior de nuestro método de análisis por niveles, llegamos a la interfaz verdaderamente humana del sistema de archivos, compuesta por nombres que definen el contenido y ayudan a identificarlo con un simple golpe de vista. Si el usuario tuviera que localizar sus archivos mediante tablas de números en vez de un título más o menos autoexplicativo como “Presupuesto\_Zona\_Norte.xls”, en una carpeta llamada “Departamento\_Comercial”, aún estaríamos bastante lejos de ver realizada la visión de Bill Gates de un ordenador en cada hogar. El sexto nivel de nuestro modelo de capas está compuesto por nombres de archivos y directorios. Una vez más los objetos de evidencia que el investigador pueda hallar trabajando en esta capa del modelo dependerán de las características concretas del sistema de archivos. Todos los nombres de archivo incluyen punteros a las estructuras de metadatos correspondientes.

### 3.3.7 Nivel 7: journaling

El *journaling* es un sistema de registro de transacciones atómicas cuya finalidad consiste en verificar que todas las operaciones de actualización de los metadatos y otras estructuras de control del sistema de archivos se llevan a cabo de manera correcta y completa. Propiamente no se trata de un nivel, sino de un conjunto de características que no todos los sistemas de archivos poseen. Sin embargo, la importancia –o mejor dicho el peligro– que el *journaling* representa para la investigación forense justifica su inclusión en el modelo de capas. Vamos a explicarlo detenidamente.

Un sistema de *journaling* funciona a base de archivos auxiliares en los que se va anotando de manera provisional el estado de una transacción ejecutada por el sistema de archivos (abrir, copiar, modificar o borrar un archivo). La transacción es incorporada a los metadatos del sistema de archivos únicamente si finaliza con éxito, tras lo cual queda anotada en el archivo auxiliar. Cuando un sistema operativo monta una partición con *journaling*, en primer lugar verifica el estado de las transacciones atómicas. Si encuentra alguna incongruencia, rectificará todos los cambios correspondientes en el sistema de archivos y volverá a dejar todo como estaba antes de la transacción incompleta. Esto hace posible que el sistema de archivos se monte con normalidad sin que sea necesario repararlo con herramientas como chkdsk.exe (Windows) o fsck (Linux).

El *journaling* constituye sin lugar a dudas un gran avance de ingeniería informática, pero es también el peor enemigo del investigador forense, no tanto por las alteraciones que pueda causar en el soporte –pues casi siempre se trata de simple información administrativa para el funcionamiento del sistema– como por sus consecuencias legales. Aunque sus efectos son mínimamente intrusivos, el

montaje automático de una partición con *journaling* modifica datos, con lo cual el *hash* de un soporte de datos montado con posterioridad a su adquisición forense no volverá a coincidir con el de la imagen realizada originalmente. Y si esto sucede, adiós a la cadena de custodia y al valor de la evidencia ante los tribunales. Por esta razón todas las copias a bajo nivel deben realizarse a través de bloqueadores como los de la figura 3-1 o mediante un software que excluya el montaje automático de particiones en modo de escritura. Se ha de tener cuidado cuando después de haber estado trabajando algún tiempo con soportes FAT (que carecen de *journaling*) se hubiera de adquirir una partición NTFS, y sobre todo durante el examen de discos de procedencia desconocida. Ante la duda, el único modo seguro de evitar los estragos del *journaling* es mediante el uso de bloqueadores de escritura.

### 3.4 RECUPERACIÓN DE ARCHIVOS BORRADOS

Los ordenadores son máquinas diseñadas para gestionar información mediante procesos que se ejecutan en segundo plano y escapan al control del usuario. Este se limita a solicitar a la máquina sus documentos, trabajar con ellos, guardarlos y –si es precavido– hacer así mismo las necesarias copias de seguridad. A no ser que disponga de conocimientos de Tecnologías de la Información no será consciente de la intensidad y el denuedo con que el sistema operativo trabaja en la sombra para hacerle la vida más fácil: paginación de memoria, creación de archivos temporales, mantenimiento de la papelera de Windows, gestión de historiales de Internet, elaboración de archivos de *prefetching* para acelerar el arranque del ordenador, almacenamiento de *cookies*, asociación de extensiones de archivo a los programas de uso habitual y muchas otras operaciones, ejecutadas casi todas ellas en segundo plano. Muy pocos usuarios saben que los archivos borrados pueden volver a la vida. Y de ellos, la mayor parte se llevarían una sorpresa si se les dijera que este rescate es perfectamente posible incluso después de haber formateado un disco duro.



Figura 3.8. ¿Está totalmente seguro de haber eliminado sus datos?

#### 3.4.1 Dinámica del borrado de archivos

Cuando el usuario borra un archivo este no desaparece del disco duro. El sistema simplemente libera los sectores ocupados por dicho archivo y los señala como disponibles para guardar otros datos. Mientras no se escriba nada encima, el contenido del archivo seguirá estando allí y podrá recuperarse con herramientas automatizadas o manualmente por medio de un editor hexadecimal. Los sistemas operativos no hacen esto para facilitarle la vida al investigador, sino por razones de economía. Aunque parezca un contrasentido, eliminar datos cuesta tanto trabajo como guardarlos, e incluso más: hay que cubrir con ceros o caracteres aleatorios los sectores que ocupaba el archivo, y esto, además de llevar tiempo, implica un consumo innecesario de recursos del sistema –por no hablar del esfuerzo que supondría la búsqueda de partes duplicadas del archivo en directorios temporales, archivos de paginación o particiones de intercambio–. De modo que para evitar una penalización en el rendimiento, los creadores de los primeros sistemas operativos decidieron que el borrado de archivos fuese aparente y que solo quedara completo en el momento de reasignar los bloques de datos disponibles a nuevos archivos. No deje de pensar en ello cada vez que deseche un soporte de datos usado (figura 3.8), sobre todo si lo ha utilizado para guardar información confidencial o informes referentes a casos antiguos. Podría llevarse una sorpresa bastante desagradable.

Cada sistema operativo tiene su modo característico de “dar de baja” un archivo. Las versiones de Windows basadas en NTFS lo marcan como borrado en la Mft sin tocar el nombre, los metadatos ni el contenido. FAT sustituye por un guion bajo (\_) el primer carácter del nombre de archivo. En las particiones ext2 y ext3 de Linux se elimina el nombre completo del archivo, pero sus metadatos y los punteros a los bloques de datos en los que se encuentra guardado el contenido siguen estando intactos en la tabla de *inodes*. El ejemplo equivalente en la economía de la información anterior a la invención del proceso digital de datos sería un bibliotecario que tacha el título del libro y el nombre del autor, dejando la ficha del archivo con su firma dentro del archivo.

Por lo general y en la mayor parte de los sistemas de archivos formatear una partición consiste en inicializar sus estructuras de control: Mft, FAT, tabla de *inodes*, mapas de bits y demás. Esto no afectará al contenido de la mayor parte de los archivos, cuyos datos –imágenes, texto, bases de datos, código ejecutable, etc.– seguirán estando en la partición. Aunque ya no exista ninguna referencia a ellos en el nuevo sistema de archivos, por haber sido reinicializadas sus entradas de directorio, las FAT e *inodes*, todavía pueden ser recuperados mediante el empleo de herramientas especiales. Es como si al bibliotecario de nuestro ejemplo le limpiaran de fichas todos los cajones del archivo dejando los libros en las estanterías.

### 3.4.2 Sector/, cluster/ y file slack

Aun después de haber creado archivos nuevos y guardarlos sobre el espacio liberado en el disco duro, en ocasiones todavía será posible hallar restos de archivos existentes con anterioridad. Esto se debe a la forma en que el sistema de archivos gestiona el almacenamiento de datos. Los discos duros se dividen en sectores, y estos son agrupados posteriormente en bloques de datos o *clusters*, que el sistema operativo maneja como unidades indivisibles independientemente del número de *bytes* que contengan. En el momento de formatear la partición para instalar sobre ella un sistema de archivos determinado, el sistema operativo establece un número de sectores fijo para la longitud de cada bloque de datos o *cluster*. Cualquier archivo que se guarde en la partición no ocupará su espacio exacto en *bytes*, sino que por necesidades técnicas el sistema le asignará un número entero de bloques de datos. Si nuestro archivo tiene un tamaño de 2.050 *bytes*, a la hora de guardarlo en disco el sistema operativo le asignará dos bloques de datos, ocho sectores o 4.096 *bytes*. El hueco sobrante hasta ese límite de 4.096 *bytes* se denomina *file slack* (*slack* o resto de archivo) y está compuesto de dos partes: un espacio sobrante hasta el final del sector que ha quedado incompleto o “sector slack”, también denominado Slack de RAM MSDOS, y otro que comprende lo que queda del espacio no utilizado hasta el final del segundo bloque de datos, llamado “cluster slack” o Slack de disco duro.

En general se verifica: Slack de archivo = Slack RAM MSDOS + Slack de disco duro. El lector lo comprenderá con facilidad si se fija en el esquema siguiente (figura 3.9). En la representación gráfica cada bloque de datos consta de cuatro sectores, y la zona sombreada corresponde a la extensión real del archivo.

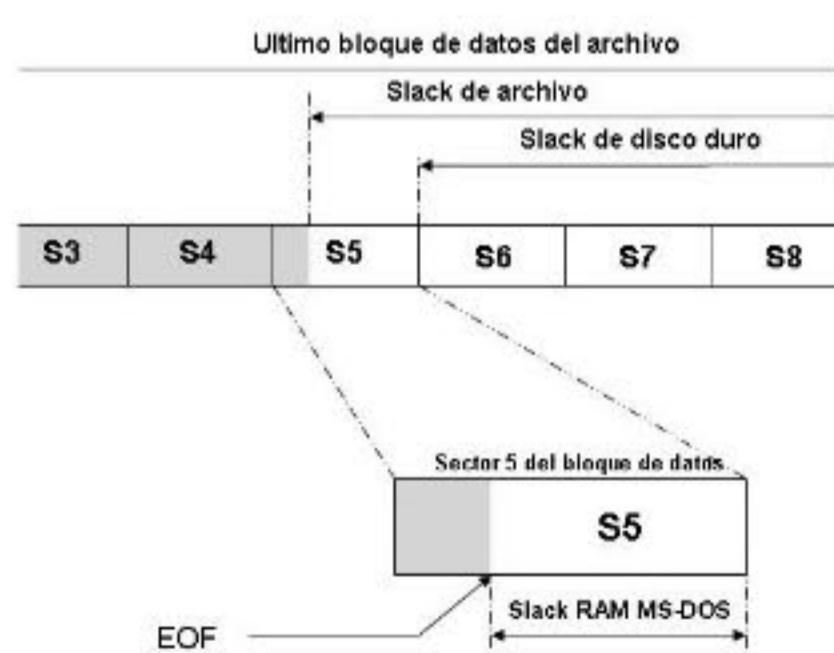


Figura 3.9. Representación esquemática del Slack

En estos huecos residuales el investigador puede encontrar datos procedentes de archivos anteriores: documentos de texto, mensajes de correo electrónico, código ejecutable, etc. Cuando los ordenadores funcionaban con sistemas operativos MSDOS, el *slack* constituía una amenaza para la seguridad. Entonces como hoy, y por conveniencias de diseño, los sistemas operativos no escribían al disco cantidades aleatorias de datos, sino sectores completos. En caso de que el archivo que quisiéramos guardar no fuese lo suficientemente grande para llegar al final del último sector de 512 *bytes*, el hueco restante se completaba con cualquier cosa que viniera a continuación del archivo en la RAM. Podía tratarse de código ejecutable o mensajes de alerta del sistema, pero también de información confidencial y contraseñas. Los sistemas Unix no presentan este problema, ya que el resto del espacio hasta el final del sector se llena con caracteres aleatorios o ceros.

### 3.5 ANÁLISIS DE UNA IMAGEN FORENSE CON TSK

El estudio de un soporte de datos se puede abordar de diversos modos. Podemos utilizar sofisticadas y potentes *suites* de investigación forense como EnCase de Guidance Software o FTK de Access Data, con interfaces gráficas al estilo Windows. También lo podemos hacer a través de un editor hexadecimal, leyendo directamente sectores de arranque, tablas de particiones, entradas de directorio, *inodes* y bloques de datos. La diferencia viene a ser más o menos como la que habría entre utilizar una excavadora o pico y pala, un dilema que normalmente se resuelve en función de las disponibilidades económicas más que por la necesidad de conservar habilidades tradicionales entre la fuerza de trabajo.

Entre ambos extremos —sencillez y ahorro de tiempo totalmente automatizados a un precio considerable frente a una ardua labor de aprendizaje y análisis a bajo nivel y poco menos que manual con herramientas difíciles de manejar pero gratuitas— disponemos de The Sleuth Kit, un conjunto de herramientas de código libre creadas por Brian Carrier y disponibles en su página web: [www.sleuthkit.org](http://www.sleuthkit.org). Si el lector es usuario de Linux puede seguir las explicaciones siguientes replicándolas en su ordenador, y cuando lo haya hecho entenderá perfectamente el procedimiento básico de un análisis *post mortem* no solo con TSK, sino con cualquier otra herramienta utilizada para el mismo fin, ya sea propietaria o de código libre.

TSK es un perfeccionamiento de TCK (*The Coroner Toolkit*), colección de programas desarrollados por Dan Farmer y Wietse Venema en 1999 para el análisis

*post mortem* de sistemas Unix. The Sleuth Kit (<http://www.sleuthkit.org>) optimiza y amplía el funcionamiento de TCK con características nuevas y capacidad para analizar otros sistemas de archivos además de los basados en Unix. Las versiones recientes admiten imágenes en los formatos EWF y AFF procedentes de adquisiciones realizadas con EnCase y otras herramientas comerciales.

### 3.5.1 Componentes de TSK

The Sleuth Kit está compuesto por una veintena de herramientas que funcionan en línea de comando. Actualmente TSK se halla incluido en todas las distribuciones Linux especializadas en seguridad informática o reparación de sistemas, como Backtrack, Knoppix o SystemRescueCD. También está disponible en los repositorios de Ubuntu y otras distribuciones orientadas al usuario. Existe la posibilidad de compilarlo para sistemas OSX y una versión para Windows que funciona bajo el entorno Cygnus de emulación Unix. Si deseamos disponer de la última versión con las características más recientes podemos descargar el código fuente de la página web del desarrollador para compilarlo y hacer una instalación por nuestra cuenta.

Brian Carrier –creador del modelo de capas al que antes se ha hecho referencia– clasifica las herramientas de TSK por el tipo de tareas a las que están dedicadas de acuerdo con los diferentes niveles de funcionalidad. Cada herramienta lleva asignado un nombre que indica su tarea en el modelo de capas. Así, por ejemplo, tendríamos los siguientes grupos de herramientas:

- “mm-”: herramientas que trabajan con volúmenes (*media management*): mmstat, mmls, mmcatt.
- “fs-”: herramientas que sirven para el análisis de sistemas de archivos: fsstat.
- “blk-”: herramientas que funcionan en el plano de los bloques de datos: blkls, blkcat, blkstat, blkcalc.
- “i-”: herramientas para la investigación de metadatos (*inodes*): icat, ifind, istat, ils.
- “f-”: herramientas características del nivel de nombres de archivos: fls, ffind.

TSK incluye otras herramientas para el nivel de *journaling* (jcat, jls), análisis de formato de imágenes (img\_stat, img\_cat) y un pequeño grupo de aplicaciones automatizadas para tareas complejas:

- tsk\_comparedir: compara una jerarquía local de directorios con otra presente en una imagen. Esto puede resultar útil para detectar *rootkits*.
- tsk\_gettimes: extracción de todos los datos temporales para construir una línea de tiempo.
- tsk\_loaddb: transfiere todos los metadatos de una imagen a una base de datos SQLite, lo cual resulta de gran ayuda cuando se quiere estudiar la evidencia con herramientas desarrolladas para otros sistemas operativos o creadas con algún lenguaje de programación específico.
- tsk\_recover: extrae a un directorio los archivos existentes en una imagen de disco.
- disk\_sreset: elimina temporalmente una HPA (*Host Protected Area*) en caso de que la misma exista. De este modo se podrá acceder a datos guardados en una parte normalmente inaccesible del soporte de datos. Una vez realizada la adquisición y reseteado el sistema, la HPA volverá a aparecer.
- disk\_stat: indica la presencia de una HPA.
- mactime: utiliza la salida de las herramientas fls e ils para crear una línea de tiempo descriptiva de la actividad realizada con los archivos.
- sorter: selecciona archivos por el tipo de los mismos, comprobando sus extensiones y la correspondencia de *hashes* con bases de datos de archivos conocidos.
- sigfind: busca valores binarios con un *offset* o desplazamiento determinado. Muy útil para recuperar estructuras de datos.

TSK también funciona a través de un interfaz gráfico llamado Autopsy, el cual facilita el manejo de las herramientas, permite unificar la gestión de casos de investigación y simplifica la realización de informes forenses (figura 3.10). Para más información sobre TSK y Autopsy el lector puede recurrir a las páginas del manual en Linux, la documentación del software y otros materiales disponibles en la página web de Brian Carrier.

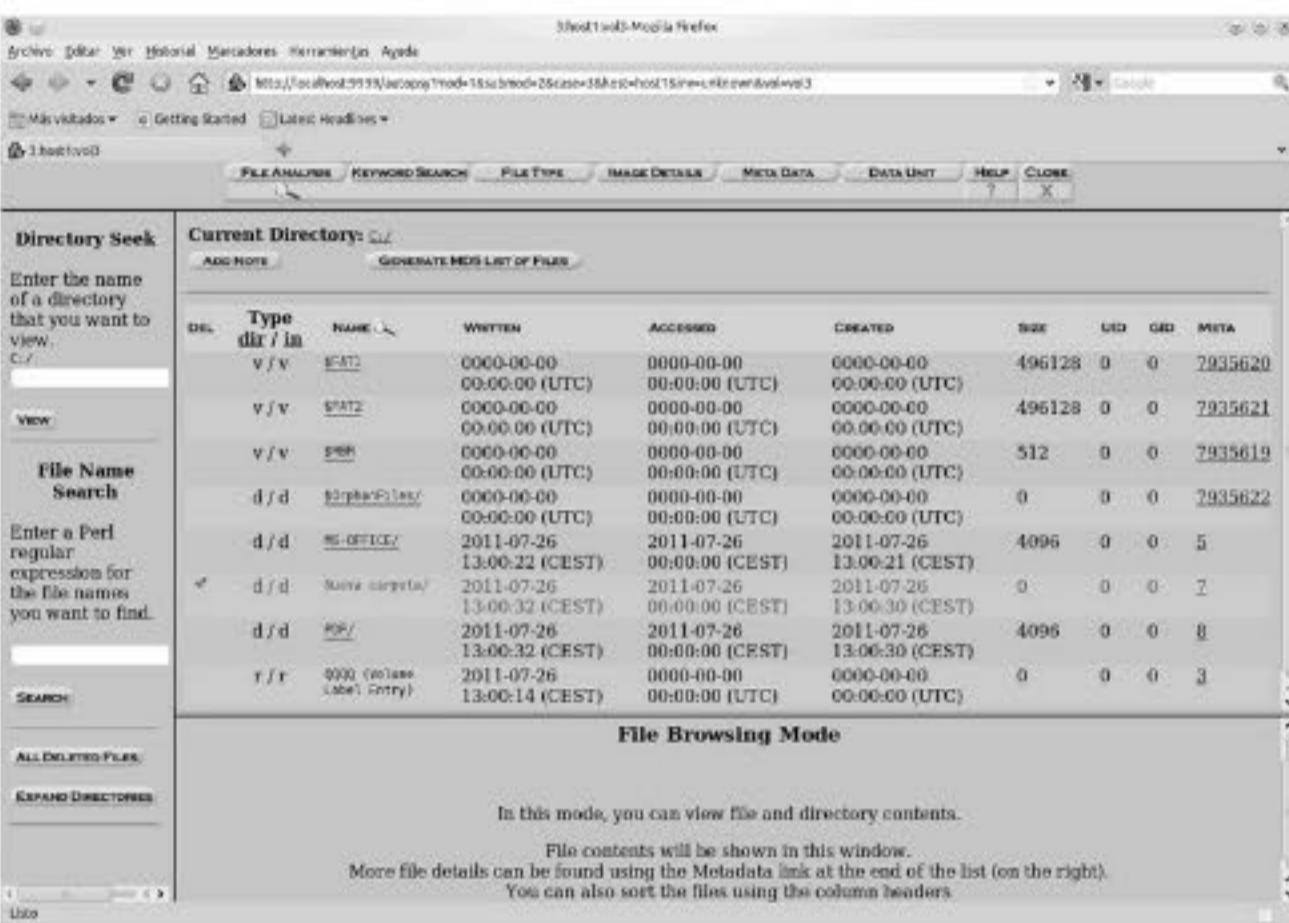


Figura 3.10. Autopsy

### 3.5.2 Adquisición de un soporte de datos

Supongamos que tenemos un soporte de datos sospechoso, en este caso una llave USB de 256 MB. Si nos hemos decidido por un medio de capacidad tan reducida se debe únicamente a conveniencias de comodidad y de tiempo: lo que vamos a hacer funcionaría perfectamente con un disco duro de gran tamaño. En primer lugar hacemos una copia a bajo nivel por cualquiera de los métodos mencionados en apartados anteriores. Abrimos una consola de texto –¡no olvide refrescar sus nociones de *bash*! ¡Y tampoco andarse con cuidado cuando tenga que trabajar como superusuario o *root*! En sistemas basados en Ubuntu Linux utilizará con frecuencia el comando *sudo* para adquirir privilegios de *root*; los resultados son los mismos– y tecleamos lo siguiente:

```
sudo dd if=/dev/sdb of=imagen_pendrive.dd
```

Haciendo esto conseguiremos crear en el disco duro de nuestra estación de trabajo forense un archivo que se corresponde bit por bit con el soporte de datos de origen. Podemos verificarlo si queremos comparando los *hashes*. La imagen contendrá, además de los archivos grabados en la llave USB, el sector de arranque y la tabla de particiones, archivos borrados y una cantidad de espacio no asignado en el que aún podría haber restos de archivos anteriores al formateado más reciente del soporte. Todavía no sabemos lo que hay dentro. Ni siquiera podemos decir si la llave USB ha sido sobreescrita con ceros o datos aleatorios, ni siquiera para qué

sistema de archivos está formateada. Todo esto lo vamos a averiguar con la ayuda de TSK. Tenga en cuenta que esto es solamente un experimento educativo. En condiciones reales nos veríamos obligados a realizar la adquisición a través de un bloqueador de escritura.

### 3.5.3 Instalación de TSK

Si es usuario de Ubuntu, en la misma consola de texto deberá teclear:

```
sudo apt-get install sleuthkit
```

TSK está disponible en todos los repositorios de Ubuntu. Pero si utiliza otra distribución, un entorno Cygnus bajo Windows o simplemente necesita la versión más reciente del software, puede descargar el código fuente desde la página del desarrollador en Sourceforge (<http://sourceforge.net/projects/sleuthkit/files/sleuthkit/>). TSK viene dentro de un archivo comprimido con denominación similar a esta: sleuthkit-X.X.X.tar.gz (donde X.X.X son los números correspondientes a la versión de TSK). Descomprimalo, descienda un nivel hasta el directorio donde se encuentra alojado el código y ejecute la rutina habitual de compilación e instalación bajo Linux:

```
tar -xfv sleuthkit-X.X.X.tar.gz
cd sleuthkit-X.X.X
./configure
make
make install
```

Las herramientas de desarrollo, incluido el compilador g++, deben estar previamente instaladas. Consulte las páginas man para saber más sobre TSK y su funcionamiento.

### 3.5.4 Análisis de la imagen

Lo que vamos a ver es tan solo una parte de la gama de herramientas y opciones de TSK. Hay muchas cosas que se pueden hacer con este software, como por ejemplo rastrear mediante el comando *ifind* el número de *inode* correspondiente a una entrada de directorio, recuperar particiones borradas con *sigfind*, etc. Aquí solo se va a exponer la operativa básica de análisis con TSK. Lo primero que interesa saber es cómo está estructurada nuestra imagen, si contiene algún tipo de sistema de archivos y, en caso de que así sea, de qué tipo de sistema de archivos se trata.

```
mmls imagen_pendrive.dd
```

El resultado:

DOS Partition Table				
Offset Sector: 0				
Units are in 512-byte sectors				
Slot	Start	End	Length	Description
00: Meta	00000000000	00000000000	00000000001	Primary Table (#0)
01: ----	00000000000	0000002047	0000002048	Unallocated
02: 00:00	0000002048	0000206847	0000204800	NTFS (0x07)
03: 00:01	0000206848	0614402047	0614195200	NTFS (0x07)
04: 00:02	0614402048	0809713663	0195311616	Linux (0x83)
05: ----	0809713664	0809715711	0000002048	Unallocated
06: Meta	0809715710	1953523711	1143808002	DOS Extended (0x05)
07: Meta	0809715710	0809715710	0000000001	Extended Table (#1)
08: 01:00	0809715712	1949618175	1139902464	Linux (0x83)
09: Meta	1949618176	1953523711	0003905536	DOS Extended (0x05)
10: Meta	1949618176	1949618176	0000000001	Extended Table (#2)
11: ----	1949618176	1949620223	0000002048	Unallocated
12: 02:00	1949620224	1953523711	0003903488	Linux Swap / Solaris x86 (0x82)
13: ----	1953523712	1953525167	0000001456	Unallocated

La salida de mmls es una tabla en la que se indica cómo está estructurado el soporte de datos. Cada registro (*Slot*) corresponde a un área funcional del disco. En él se indican sector inicial (*Start*), sector final (*End*), longitud del área en sectores de 512 bytes (*Length*) y tipo del sistema de archivos (*Description*). Nuestro soporte es, como era de esperar, la típica llave USB formateada con un sistema de archivos FAT32. Esto nos da una pista relativa a la posibilidad de que existan datos o fragmentos de archivos anteriores, ya que cuando los medios de esta capacidad aún se hallaban disponibles en el mercado, lo normal es que vinieran preformateados de fábrica con un sistema de archivos FAT16. Probablemente la llave USB ha sido formateada varias veces.

El primer sector contiene la tabla primaria de particiones. La partición FAT32 propiamente dicha, con un tamaño de 497.952 (aprox. 256 MB) no viene inmediatamente a continuación, sino que comienza tras un intervalo de 63 sectores. Al final del medio de almacenamiento, y después de la partición señalada con el tipo Win95 FAT32 (0x0B), hay un espacio sin asignar de 5.793 sectores (aprox. 2,8 MB).

Lo que hemos visto no sirve para transmitir una noción adecuada de la potencia analítica de TSK. Si el lector utiliza una máquina provista de arranque dual Linux/Windows, puede dirigir mmls contra su propio disco duro:

```
sudo mmls /dev/sda
```

En una configuración de arranque dual típica, por ejemplo la que tiene el autor de este libro en su estación de trabajo, la salida de mmls podría ser algo parecido a esto:

DOS Partition Table				
Offset Sector: 0				
Units are in 512-byte sectors				

Slot	Start	End	Length	Description
00: Meta	00000000000	00000000000	00000000001	Primary Table (#0)
01: ----	00000000000	0000002047	0000002048	Unallocated
02: 00:00	0000002048	0000206847	0000204800	NTFS (0x07)
03: 00:01	0000206848	0614402047	0614195200	NTFS (0x07)
04: 00:02	0614402048	0809713663	0195311616	Linux (0x83)
05: ----	0809713664	0809715711	0000002048	Unallocated
06: Meta	0809715710	1953523711	1143808002	DOS Extended (0x05)
07: Meta	0809715710	0809715710	0000000001	Extended Table (#1)
08: 01:00	0809715712	1949618175	1139902464	Linux (0x83)
09: Meta	1949618176	1953523711	0003905536	DOS Extended (0x05)
10: Meta	1949618176	1949618176	0000000001	Extended Table (#2)
11: ----	1949618176	1949620223	0000002048	Unallocated
12: 02:00	1949620224	1953523711	0003903488	Linux Swap / Solaris x86 (0x82)
13: ----	1953523712	1953525167	0000001456	Unallocated

Esta es la arquitectura típica de un disco duro formateado conforme al esquema tradicional MSDOS. Obsérvense las dos particiones NTFS correspondientes a la instalación normal de Windows Vista/7 (*Slots* 02 y 03) y la partición Linux en la tercera entrada de la tabla (*Slot* 04). A continuación viene la partición extendida, y dentro de ella las particiones lógicas necesarias para acomodar otra partición Linux (*Slot* 06) y una zona de intercambio o *swap* (*Slot* 09). Nótese el grado de detalle con el que mmls disecciona el disco, incluyendo las tablas correspondientes (*Slots* 07 y 10) y los espacios sin utilizar (*Slots* 1, 5 y 11). Si un sospechoso quisiera ocultar información en estas zonas del disco resultaría imposible llegar hasta esos datos con las utilidades de gestión de un sistema en funcionamiento, como por ejemplo fdisk. Sin embargo, dar con ella es relativamente fácil cuando se dispone de mmls para localizar espacios sin utilizar y de dd para extraerlos en forma de un archivo de imagen.

### 3.5.5 Análisis del sistema de archivos

La herramienta mmls pertenece al nivel lógico de volúmenes y particiones descrito en el apartado precedente de este capítulo: detecta los tipos de particiones para los que está formateado el soporte, pero no nos dice si existe efectivamente un sistema de archivos FAT, NTFS, Linux o de cualquier otra clase. Para averiguarlo tenemos que utilizar herramientas correspondientes al nivel de sistema de archivos. Volviendo a la imagen de la llave USB tecleamos en la consola de texto (cuando se trabaja sobre la imagen adquirida no son necesarios privilegios de superusuario y por ello no es preciso recurrir al comando sudo):

```
fsstat imagen0.dd
```

Y entonces... no sucede nada. Lo único que recibimos en pantalla es un mensaje de error: “*Cannot determine file system type*”. Resulta un poco frustrante. La salida de mmls nos indica que el volumen contiene una partición FAT. ¿Qué ha

sucedido? Examinémoslo con atención. Según mmls, entre el primer sector (tabla primaria de particiones del soporte) y el comienzo de la partición Win95 FAT32 existe un espacio sin asignar con una longitud total de 63 sectores. La herramienta TSK encargada de analizar sistemas de archivos no tiene ningún modo de saberlo. Lo que ha hecho simplemente es buscar estructuras de datos que tengan sentido para el código con que está programada en una zona del medio a la que accede por defecto al ejecutarse sin parámetros desde la consola de texto. Al no encontrar nada que pueda interpretar correctamente devuelve un mensaje de error. Pero si le indicamos que omita esos 63 sectores mediante un salto (*offset*), de este modo:

```
fsstat -o 63 imagen0.dd
```

entonces funcionará como es debido, entregando una salida parecida a esto:

```
FILE SYSTEM INFORMATION
-----
File System Type: FAT32
OEM Name: MSDOS5.0
Volume ID: 0xb4309d75
Volume Label (Boot Sector): NO NAME
Volume Label (Root Directory): QQQQ
File System Type Label: FAT32
Next Free Sector (FS Info): 5044
Free Sector Count (FS Info): 492904
Sectors before file system: 63
File System Layout (in sectors)
Total Range: 0 - 497951
* Reserved: 0 - 37
** Boot Sector: 0
** FS Info Sector: 1
** Backup Boot Sector: 6
* FAT 0: 38 - 1006
* FAT 1: 1007 - 1975
* Data Area: 1976 - 497951
** Cluster Area: 1976 - 497951
*** Root Directory: 1976 - 1979

METADATA INFORMATION
-----
Range: 2 - 7935622
Root Directory: 2

CONTENT INFORMATION
-----
Sector Size: 512
Cluster Size: 2048
Total Cluster Range: 2 - 123995
```

No solo hemos logrado averiguar que el soporte sospechoso contiene efectivamente una partición FAT32. Ahora conocemos también detalles de gran interés para nuestra investigación, sobre todo para el caso de que sea necesario recuperar archivos borrados: identificador del volumen, rangos de almacenamiento de datos, *clusters* o bloques de datos disponibles, ubicación de las dos tablas de localización de archivos (FAT), posición del directorio raíz y el tamaño del bloque de datos (en este caso 4 sectores o 2.048 bytes). Hemos cortado la salida del comando para poder explicarla por partes. El *output* restante describe el contenido de la FAT con la ubicación de cada archivo dentro del soporte de datos.

```
.....
.....
FAT CONTENTS (in sectors)
-----
1976-1979 (4) -> EOF
1980-1983 (4) -> 4964
1984-1987 (4) -> 35156
1988-2059 (72) -> EOF
2060-2311 (252) -> EOF
2312-3195 (884) -> EOF
3196-3259 (64) -> EOF
3260-3311 (52) -> EOF
3312-3371 (60) -> EOF
3372-3443 (72) -> EOF
3444-3515 (72) -> EOF
3516-3567 (52) -> EOF
3568-3791 (224) -> EOF
3792-3851 (60) -> EOF
3852-4051 (200) -> EOF
4052-4963 (912) -> EOF
4964-4967 (4) -> EOF
4968-5043 (76) -> EOF
35156-35159 (4) -> EOF
```

Los números entre paréntesis indican la longitud del archivo en *clusters*. EOF (*End Of File*) quiere decir que el archivo está completo y termina ahí. Si en lugar de haber una marca EOF en la entrada de la FAT nos encontramos con un número de sector, eso significa que el archivo dispone de nuevos contenidos en la nueva posición y por tanto está fragmentado. Este es por ejemplo el caso en la tercera línea, donde tenemos un archivo que ocupa 4 bloques de datos desde el *cluster* 1.984 al 1.987 y continúa en el *cluster* 35.156, ocupando otros 4. El espacio en disco de dicho archivo es de 8 bloques de 2.048 bytes o 16 Kilobytes. Sin embargo, y de acuerdo con lo que se explicó anteriormente al hablar del *slack*, ese no tiene por qué ser el tamaño real del archivo. Este podría tener 16 kilobytes completos o tan solo algo más de 14, dependiendo de la ocupación efectiva del último bloque de datos asignado. Del mismo modo en dicho bloque de datos podría existir información perteneciente a archivos borrados con anterioridad.

### 3.5.6 Listado de archivos

Las herramientas de TSK utilizadas hasta el momento sugieren que este soporte de datos contiene algunos archivos, la mayoría de ellos de reducida extensión. Queremos saber algo más sobre estos elementos de evidencia potenciales, por ejemplo nombres y extensiones. También nos gustaría averiguar si hay archivos borrados. La herramienta fsstat omite la información relativa a entradas FAT con el número hexadecimal 0000, indicador de que el bloque de datos correspondiente está disponible. Allí podría haber datos pertenecientes a otros archivos eliminados anteriormente por el usuario.

Ahora nos encontramos en el nivel número seis del modelo: nombre de archivos. La herramienta que necesitamos es fls. Al utilizarla incluiremos también el salto de 63 sectores hasta el comienzo de la partición FAT:

```
fls -o 63 imagen0.dd
```

Resultado:

```
r/r 3: QQQQ (Volume Label Entry)
d/d 5: MS-OFFICE
d/d * 7: Nueva carpeta
d/d 8: PDF
v/v 7935619: $MBR
v/v 7935620: $FAT1
v/v 7935621: $FAT2
d/d 7935622: $OrphanFiles
```

Originalmente TSK fue diseñado para el análisis forense de sistemas Unix. Por esta razón emplea números de *inode* para los archivos y otras estructuras de datos, aunque se trate de particiones NTFS o FAT. Cada número de *inode* equivale más o menos a una entrada en la Mft. Sin embargo, en este nivel lo que interesa son nombres de archivos y carpetas (indicadas por d/d).

Obsérvese que aún no hemos montado en ningún ordenador el soporte de datos ni la imagen del mismo. Tan solo estamos examinando su contenido con una máquina digital de rayos X. En el nivel de interfaz humana que constituyen los nombres de archivo encontramos una carpeta llamada "MS-OFFICE" con el número de *inode* 5. También hay algo que llama la atención: una carpeta con el nombre "Nueva Carpeta" que figura como borrada. Probablemente el usuario la creó con el explorador de Windows, renombrándola acto seguido a "PDF". El asterisco delante del número de *inode* significa que un archivo o una carpeta han sido borrados. Adviértase que los números de *inode* de las dos carpetas son consecutivos.

Por defecto fls muestra únicamente los archivos del directorio raíz. Para ver el contenido de carpetas y subdirectorios utilizamos la opción -r:

```
fls -o 63 -r imagen0.dd
```

Y con esto lograremos que nuestra máquina de rayos X llegue un poco más hondo en el sistema de archivos, es decir, en el esqueleto de la imagen que estamos examinando:

```
r/r 3: QQQQ (Volume Label Entry)
d/d 5: MS-OFFICE
+ r/r 71: GonSoto_Adopcion.doc
+ r/r 80: 60-3-BMLVS_GZ_E90023_6_00-01_KA_2011_RFP_Concurso_Municiones.doc
+ r/r 87: 60-7-BMLVS_TL_40X53_TP_T_Concurso_Municiones.doc
+ r/r 90: INFORME CONF.doc
+ r/r 94: Registro_Mercantil_Essen_FIDAB.doc
+ r/r 97: Papeles_Directivos.doc
+ r/r 101: KRAWA_Registro_Mercantil.doc
+ r/r 104: Ariane_Friedberger.doc
+ r/r 107: Magister_doc_spanisch.doc
+ r/r 111: Deloitte_Extracto_RegMerc_Zug.doc
+ r/r 115: Escrituras_Notario_Tejerina.doc
+ r/r 118: Notario_WAMS.doc
+ r/r 126: 60-7-BMLVS_TL_40X53_HE_SD_Concurso_Ministerio_Defensa.doc
+ r/r 47814: 60-2-BMLVS_40X53_AAB_Concurso_Ministerio_Defensa.doc
d/d * 7: Nueva carpeta
d/d 8: PDF
+ r/r * 136: 20100831_E-Steiermark_Angebot.pdf
+ r/r * 140: 20100831_Ecgas_Angebot.pdf
+ r/r * 146: 30199ORDEESCAF_Anerkennung_Spezifikationen_SBB20110705.pdf
+ r/r * 151: 11-0325-ES081-FRED_ROLF_OSTREICH-alemán.pdf
+ r/r * 158: Autorización_3521117_Heparine_Natrium_25000_IE_Vial_5_ml_Belgium.pdf
+ r/r * 161: Autorización_Suiza.pdf
+ r/r * 164: 60-2-BMLVS_40X53_AAB.pdf
+ r/r * 169: 60-3-BMLVS_GZ_E90023_6_00-01_KA_2011_RFP.pdf
+ r/r * 173: 60-5-BMLVS_TL_40X53_HE_SD_T.pdf
+ r/r * 177: 60-7-BMLVS_TL_40X53_TP_T.pdf
+ r/r * 180: UNI ALEMANIA IHK.pdf
+ r/r * 183: INFORME CONF2.PDF
+ r/r * 186: BIOGAS_-_Aval.PDF
+ r/r * 191: ALCIRA-Notificacion_Alemania_110211.pdf
+ r/r * 530886: BILDUNGSWERK_DER_HESSISCHEN_WIRTSCHAFT.pdf
+ r/r * 530890: TRADUCCION_JURADA_FELDBERGSCHULE.pdf
+ r/r * 530894: ILS_247_text_in_German.pdf
+ r/r * 530897: S25C-410121514480.pdf
+ r/r * 530902: 1011-4895_Scan_Urteil_7_O_151_09_v_6_Juli_2010.pdf
v/v 7935619: $MBR
v/v 7935620: $FAT1
v/v 7935621: $FAT2
d/d 7935622: $OrphanFiles
```

He aquí el auténtico poder de la interfaz humana de TSK. Ya no se trata de números ni especificaciones de ingeniería, sino de nombres de archivos, documentos, información con un significado concreto que permite incluir nuestro hallazgo en sumarios judiciales o informes de seguridad. En primer lugar una carpeta con documentos Office, que al no haber sido eliminados podremos examinar sin dificultad una vez montado el sistema de archivos. A continuación otra carpeta con documentos Acrobat PDF. Estos archivos, al contrario que los de la carpeta anterior, sí han sido borrados, como indica el asterisco que figura frente a los números de *inode*.

### 3.5.7 Recuperando archivos borrados

En las entradas de directorio aún figuran los nombres de los archivos PDF, lo cual sugiere que existe una alta probabilidad de recuperarlos, sobre todo si no están fragmentados. Nos gustaría examinar alguno de ellos. ¿Qué tal por ejemplo el que lleva asignado el número de *inode* 140? Para ello recurrimos al comando icat, correspondiente al nivel 5 (metadatos) del modelo de capas:

```
icat -o 63 imagen0.dd 140 > prueba.pdf
```

Ejecutando el comando sin más los resultados serían enviados a la salida estándar, es decir, a la pantalla, y lo único que veríamos sería un galimatías de texto y códigos de formato. Sospechamos que se trata de un documento Adobe Acrobat PDF, pero en realidad puede ser cualquier cosa. Alguien podría haber cambiado la extensión con el propósito de ocultar el contenido. De momento solo podemos hacer suposiciones. Para verificarlas redirigimos la salida a un archivo con extensión PDF. El redireccionamiento permite introducir la salida de un comando como entrada de otro. Se trata de una técnica disponible en todos los sistemas operativos basados en línea de comando, como por ejemplo Linux/Unix, el antiguo MSDOS y los interfaces de consola existentes en versiones modernas de MS-Windows y Mac OSX. El archivo generado contiene los mismos datos que el original.

Abrimos el archivo de prueba con una aplicación como Adobe Acrobat Reader, Okular o cualquier visor de documentos, y descubrimos que efectivamente se trata de un documento PDF. Gracias a TSK acabamos de recuperar nuestro primer archivo borrado. Si el lector ha seguido las explicaciones reproduciéndolas en su propio ordenador y con cualquier soporte de datos USB preparado al efecto, este ejercicio le permitirá comprender mejor la operativa de funcionamiento del software utilizado para la recuperación forense de archivos. Las *suites* comerciales como EnCase, FTK, SMART o MacForensicLab, no hacen otra cosa que aplicar de manera masiva y automatizada unas técnicas similares a las que hemos utilizado manualmente con TSK.

## 3.6 ANÁLISIS DE ARCHIVOS

Si estamos buscando elementos de evidencia, lo más probable es que al final los hallemos en el interior de un archivo. Por ello, y debido a que identificar archivos no es tan sencillo como deducir el tipo de los mismos a partir de su extensión –el cambio de extensión es una técnica de camuflaje muy utilizada por los delincuentes informáticos–, el investigador tiene que familiarizarse con los fundamentos del análisis de archivos.

A todo esto, ¿qué es un archivo informático? A efectos prácticos tan solo una secuencia de datos hexadecimales –en última instancia binarios– que puede significar cualquier cosa (texto, código ejecutable, puntos de una imagen, sonido, información encriptada, etc.). El sistema operativo procesa los archivos como una unidad, leyéndolos, copiándolos en memoria, guardándolos y borrándolos en su integridad y no por partes. En el entorno del PC de sobremesa, el archivo admite nombre (obligatorio) y extensión (opcional). Algunos sistemas operativos –por ejemplo Windows– identifican el archivo por medio de su extensión, llamando automáticamente a la aplicación de software encargada de procesarlo. Otros –Linux/Unix– lo reconocen por determinadas características de su contenido o una secuencia de caracteres específicos situada al comienzo del archivo, no siendo necesario conocer su extensión para procesarlos o llamar a las aplicaciones correspondientes.

El nombre del archivo no va incluido dentro del mismo. Tampoco la información relativa a su tamaño en *bytes*, ni las marcas de tiempo, los permisos de ejecución y otras características (invisibilidad, solo lectura, etc.). Los datos correspondientes a todas estas características se guardan externamente en las estructuras del sistema de archivos –entradas de directorios, entradas de la Mft, *inodes*–. Ello no impide que algunos archivos con formatos determinados, como documentos de Office o archivos gráficos, lleven incrustados en su interior etiquetas o metadatos que el sistema emplea para fines de administración. Más tarde se hablará de los metadatos y su interés forense.

### 3.6.1 Firmas características

Se puede saber de qué tipo es un archivo examinando su firma característica. Esta firma consiste en una secuencia de caracteres que figura al comienzo del archivo, situada de manera necesaria siempre al comienzo de un bloque de datos o *cluster*. Por ejemplo, si con un editor hexadecimal abrimos una imagen JPEG, en la primera línea hallaremos los caracteres 4a464946 (ASCII: JFIF), que es su firma o número mágico, como a veces se la llama (figura 3.11). Si cambiamos la extensión del archivo quizás logremos engañar al sistema operativo, pero seguirá siendo un archivo gráfico, y su contenido podrá ser manejado

correctamente por cualquier visor o software de retoque fotográfico. Determinar el tipo de archivo es importante en cualquier situación en la que el sospechoso pudiera disponer de un motivo para recurrir a las técnicas de ocultación.

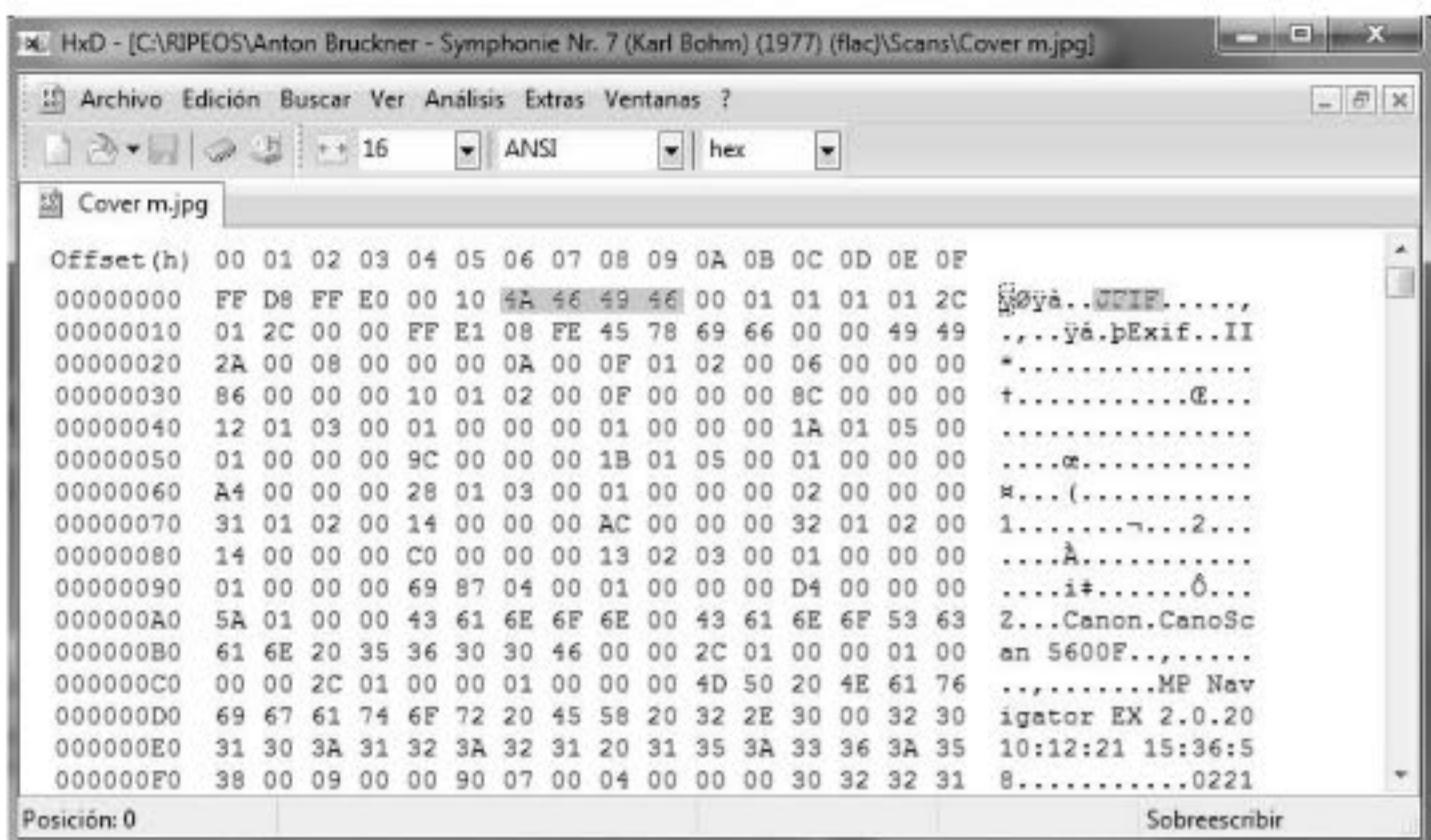


Figura 3.11. Firma característica de un archivo JPG

Siempre que el investigador se encuentra con un archivo sospechoso, ante la duda lo primero que hace es examinarlo con un editor hexadecimal, tomar nota de sus caracteres iniciales y efectuar una consulta en Google. Linux dispone de una herramienta que determina el tipo de archivo a partir de la información contenida en magic, un archivo de texto que por lo general se encuentra en el directorio /usr/share/misc. Magic es un listado con información relativa a los diferentes tipos de archivos, con sus firmas características y otros datos. Esta lista es ampliable, y el usuario la podrá complementar con firmas correspondientes a nuevos tipos de archivo con los que se vaya encontrando en el transcurso de su investigación.

Supongamos que nos han pedido analizar un archivo con extensión dll que por su tamaño (alrededor de 300 MB) parece demasiado grande para ser una librería de Windows. Para averiguarlo abrimos una consola *bash* y ejecutamos contra él nuestro comando file:

```
file re32w.dll
```

El resultado:

```
re32w.dll: RIFF (little-endian) data, AVI, 624 x 352, 23.98 fps, video: XviD, audio: MPEG-1
Layer 3 (stereo, 48000 Hz)
```

### 3.6.2 Documentos

Los documentos hallados en el transcurso de una investigación tienen interés no solo por su contenido explícito de carácter textual o gráfico, sino también por otros elementos de información que pueden extraerse de los mismos. Casi todos los formatos de documento, para finalidades de gestión o simplemente con el propósito de facilitar su localización en repositorios o Intranets, llevan en su interior unas etiquetas especiales llamadas metadatos: nombre de autor, historial de cambios, fecha de la última modificación, tiempo total trabajado, nombre del sistema, directorio de trabajo del usuario e incluso la impresora con la que se hicieron los ejemplares en papel. Algunas versiones antiguas de MS-Office llegan a incluir en los documentos de Word hasta la dirección MAC del interfaz de red del ordenador, haciendo posible la identificación exacta de la máquina en la que el archivo fue creado.

Trabajar con documentos supone un riesgo. Por lo general el investigador deberá emplear con ellos las mismas precauciones que en el análisis de código ejecutable. Algunos archivos de formato complejo, como MS-Office, admiten la inclusión de macros para automatizar tareas. Esta característica implica la posibilidad de que el documento esté contaminado con virus de macro u otras formas de código maligno capaces de afectar al funcionamiento del sistema operativo y comprometer su seguridad. Lo mismo es aplicable a archivos PDF y Flash por la posibilidad de introducir en ellos rutinas en JavaScript capaces de ejecutarse automáticamente al cargar el documento.

- **Archivos MS-Office:** este tipo de documentos viene utilizándose desde hace dos décadas y es el más abundante en todo tipo de entornos empresariales y particulares. Los archivos MS-Office (documentos de texto Word, hojas de cálculo Excel, presentaciones PowerPoint) resultan interesantes no solo por su contenido sino por los metadatos que llevan incrustados en su interior. Los metadatos constituyen una brecha importante en la seguridad corporativa de muchas empresas, pero resultan muy útiles para el investigador forense.
- **Documentos Open Office (OOXML):** se trata de un formato abierto basado en etiquetas XML para documentos editables y es estándar en las versiones nuevas de MS-Office. Un archivo con extensión docx consiste en un contenedor con diferentes *streams* (flujos de datos) en su interior. Cada uno de estos *streams* es como una especie de subarchivo con datos correspondientes a una parte del documento: texto, formato, imágenes, etc. Para desmantelar un archivo Open XML y extraer todo su contenido de manera ordenada, basta cambiar su extensión de docx a

- zip. Después el documento se puede manejar como si fuera un archivo comprimido con carpetas en su interior.
- **OpenDocument:** OpenDocument es un formato abierto desarrollado por Sun Microsystems que compite con OOXML, y que utilizan plataformas de código abierto como Open Office. Resulta reconocible por la extensión ODT. Al igual que en el caso precedente se trata de contenedores ZIP, y basta con cambiar su extensión para poder acceder a ellos con facilidad. Los metadatos se encuentran alojados en el archivo meta.xml y pueden ser examinados con cualquier visor XML.
  - **Documentos RTF:** este formato fue creado por Microsoft para facilitar la portabilidad de los documentos de unas plataformas a otras. Lo emplea de modo nativo el editor Wordpad de Windows. RTF (*Rich Text Format*) define la estructura del documento a base de etiquetas simples. Tanto el contenido del archivo como sus códigos de formato pueden ser examinados con cualquier editor de texto. Los caracteres no ASCII se representan mediante secuencias de escape y los metadatos son visibles sin necesidad de herramientas especiales.

- **Documentos PDF:** el formato PDF (*Adobe's Portable Document Format*) fue diseñado para permitir la visualización de documentos con formato propio independientemente del sistema operativo y la aplicación. Un archivo PDF creado en Windows se ve igual bajo Linux, OSX o cualquier otra plataforma. PDF se utiliza para documentos de todo tipo, desde manuales de instrucciones de máquinas hasta notas de prensa e impresos. También incluye metadatos, como por ejemplo fecha de creación del documento y software utilizado para crear el PDF a partir de un documento elaborado con un procesador de textos como MS-Word o con un programa de edición como QuarkXPress.

### 3.6.3 Archivos gráficos

En una investigación forense las imágenes, de manera análoga a los documentos de texto, no solo son importantes por lo que muestran sino también por los metadatos que contienen. Si un archivo gráfico JPG ha sido obtenido mediante una cámara digital, en su interior se podrán hallar etiquetas relativas a una gran variedad de categorías de información: fecha y hora de la toma, modelo de cámara, ajustes de disparo, longitud focal, coordenadas geográficas (esto último en teléfonos móviles de gama alta y *smartphones* equipados con cámara y GPS), software de retoque fotográfico utilizado. Existen numerosos estándares gráficos digitales. Estos son los más habituales:

- **Archivos JPEG:** los archivos JPEG o JPG (*Joint Photographic Experts Group*) deben su nombre al comité de especialistas que definió este estándar en 1992. JPEG es el formato más extendido, tanto en Internet como en electrónica de consumo (cámaras digitales, teléfonos móviles, *smartphones*). Utiliza un algoritmo de compresión con pérdida que permite capturar imágenes en archivos de tamaño variable haciendo posible un compromiso entre calidad y tamaño según las preferencias del usuario. Los archivos JPEG admiten diversos tipos de metadatos: Exif, IPTC y XMP. Otra característica del formato JPEG reside en que el método de compresión, basado en la aplicación de unas fórmulas matemáticas denominadas Transformadas de Fourier, deja en el archivo indicios de las tablas de cuantización utilizadas para reducir el volumen de datos, y que pueden ser reveladores de la marca o modelo de cámara con que la fotografía fue tomada. Así mismo en imágenes JPG resulta posible descubrir trazas de posibles intentos de manipulación mediante el empleo de métodos de análisis matemático como ELA (*Error Level Analysis*) o gradientes de luminancia.
- **GIF:** los archivos GIF (*Graphics Interchange Format*) se utilizan desde hace tres décadas para la presentación de iconos e imágenes simples. Su rango cromático –256 colores– era más que suficiente para los gráficos de entonces, pero no para las preferencias del usuario o las posibilidades del software de retoque fotográfico de nuestros días. Sin embargo tienen la ventaja de que soportan efectos de transparencia y animación. Se elaboran mediante un algoritmo de compresión sin pérdida.
- **PNG:** PNG (*Portable Network Graphics*) es un formato de compresión sin pérdida desarrollado para reemplazar a GIF en imágenes destinadas a Internet. La ausencia de un soporte para metadatos –solo admite etiquetas XMP pero es poco frecuente encontrarlas– reduce su valor informativo al contenido gráfico y a las marcas de tiempo del archivo.
- **TIFF:** el formato TIFF (*Tagged Image File Format*) se utiliza para trabajos de edición y diseño gráfico. Constituye la opción por defecto en numerosas aplicaciones para sistemas Apple OSX. Como su propio nombre indica, los archivos TIFF admiten metadatos. Existen versiones avanzadas de este formato para fotografía planimétrica y satelital mediante algoritmos de compresión sin pérdida, así como para digitalizar documentos y faxes.
- **RAW:** los formatos RAW utilizados por las cámaras digitales son por lo general desarrollos de TIFF adaptados a las necesidades de cada

fabricante. Su interés forense no reside tanto en el análisis de los mismos como en su empleo como herramienta de trabajo en peritaciones o tareas de investigación criminológica, en las que resulta necesario presentar la evidencia gráfica en un formato inalterable sin una labor adicional de procesamiento de imagen –salvo, claro está, para visualizarla en una pantalla de ordenador o imprimirla–. Al tratarse de un algoritmo de compresión sin pérdida, o más exactamente, de los propios datos en bruto captados por el CCD de la cámara, resulta difícil impugnar una foto RAW ante el tribunal con el argumento de que los datos han podido ser manipulados.

La principal desventaja es que no existe un estándar unificado para archivos RAW. Cada marca de cámara tiene su propio formato RAW adaptado a la ingeniería del sensor, y para leerlo hace falta un *driver* especial suministrado por el fabricante. Durante los últimos años se han realizado avances significativos como Adobe DNG, disponible en versiones recientes de Photoshop (CS4 y CS5). Un archivo gráfico RAW convertido a DNG mediante utilidades del fabricante podrá ser procesado por cualquier software de visualización o retoque que soporte la nueva especificación de Adobe.

### 3.6.4 Multimedia

En principio los archivos multimedia no se diferencian de los de cualquier otro tipo. Contienen datos, etiquetas y metadatos susceptibles de interpretación y análisis forense. La única dificultad que ofrecen al investigador es su enorme tamaño, lo cual supone una mayor probabilidad de fragmentación, con la consecuencia de que una recuperación de este tipo de archivos resulta problemática en caso de que hayan sido borrados.

Un archivo de vídeo digital contiene datos que al ser decodificados por un programa de visualización permiten reconstruir una secuencia de imágenes en movimiento. El flujo de datos de vídeo viene acompañado por una o varias pistas de audio, todo ello empaquetado en un contenedor que agrupa los diferentes flujos o *streams* y constituye el archivo multimedia propiamente dicho. El método utilizado para comprimir los datos de imagen y de sonido en cada uno de los *streams* se denomina *codec*. Para poder reproducir el archivo es preciso que estén instalados en el sistema los codecs de vídeo y audio correspondientes. A continuación se mencionan algunos de los estándares de compresión para vídeo y audio utilizados con más frecuencia en la actualidad:

- **MPEG-1 y MPEG-2**, este último característico de los DVD y de las transmisiones vía satélite. Ninguno de ellos incluye una cantidad apreciable de metadatos. MPEG-4 es un formato de alto rendimiento

para compresión que se emplea para codificar películas y distribuirlas en Internet a través de archivos AVI. AVI no es exactamente un formato de compresión, sino un contenedor que puede llevar dentro varios flujos de datos de vídeo y audio comprimidos con diferentes codecs. Los metadatos de un archivo AVI hacen referencia a los contenidos del archivo e incluyen gran cantidad de información técnica sobre los *streams* (dimensiones de la imagen, caudal de datos en vídeo y audio, número de cuadros por segundo, sistema de color PAL o NTSC, frecuencia de muestreo del audio, etc.). Todo ello puede resultar útil a la hora de identificar a un sospechoso por las herramientas informáticas instaladas en su ordenador.

- **WMV** (*Windows Media Video*) es un formato propietario de Microsoft para la transmisión de vídeo comprimido. Frecuentemente aparece asociado a flujos de audio WMA (*Windows Media Audio*, también propietario de Microsoft) en el interior de contenedores ASF. Apple también dispone de su propio codec propietario: **QuickTime**, utilizado en archivos multimedia con extensión MOV.
- Un desarrollo más moderno y de código abierto es el contenedor **MKV** (*Matroska Multimedia Container*), empleado en películas de alta definición gracias a su capacidad para transportar vídeo, sonido, imágenes y subtítulos.

Todos estos formatos pueden ser analizados sin problemas por las *suites* comerciales de investigación forense. En Linux existen herramientas y bibliotecas como hachoir-metadata, quicktime-utils o mkvtoolnix que nos permiten examinar la estructura interna de archivos de este tipo.

Por su parte los archivos de audio incluyen datos cuya decodificación sirve para reconstruir una señal sonora: música, mensajes de voz, grabaciones o cualquier otro tipo de material perceptible por el oído humano. Aparte de su contenido sonoro, los archivos de audio en ocasiones también incluyen metadatos que pueden ser útiles para el investigador.

- **WAV** (*Waveform Audio Format*): WAV, también denominado comúnmente archivo de onda, es un estándar de almacenamiento de datos de audio desarrollado originalmente para ordenadores de sobremesa por IBM y Microsoft. El audio se guarda en fragmentos etiquetados dentro de un contenedor RIFF, que además de sus propias etiquetas también puede llevar metadatos XMP.

- **MPEG-3** (más conocido como MP3) es el formato popular de compresión de audio más extendido. Recibe su nombre del comité facultativo que lo creó en 1993 (*Moving Picture Experts Group*), y ha sido utilizado masivamente por los programas para intercambio de archivos en red: Napster, Kazaa, eMule, BitTorrent y páginas de descargas. MP3 utiliza un formato de compresión con pérdidas que reduce el tamaño de los archivos de audio sin pérdida aparente de calidad. MP3 está basado en modelos psicofísicos de percepción del sonido y en la incapacidad del oído humano para percibir de modo simultáneo todas las frecuencias sonoras. Un archivo en bruto puede comprimirse hasta una relación de 10:1 sin que el oído note pérdidas de calidad. Esto hace posible que una canción quepa en un archivo de 4-5 MB, y que cualquier reproductor de medios de gama baja pueda contener en su interior cientos de horas de música. Los archivos MP3 incluyen metadatos en forma de etiquetas ID3v1 o ID3v2 con información relativa sobre el título de las pistas, los álbumes, el software utilizado para la extracción y compresión del sonido, etc. Estos datos pueden extraerse con herramientas de edición de medios o con cualquier programa para insertar etiquetas.
- **ASF/WMA** son formatos de audio propietarios de Microsoft. Apple dispone así mismo de M4P y M4R, frecuentes en sistemas con iTunes y diseñados para facilitar la gestión de derechos digitales (DRM).
- Finalmente, **AAC/M4A** (*Advanced Audio Coding*) es un estándar creado para reemplazar a MP3. Los *streams* de audio comprimidos con este codec se guardan por lo general en contenedores MP4.

### 3.6.5 Archivos ejecutables

A diferencia de los tipos de archivo mencionados con anterioridad, los de código ejecutable constituyen un caso especial por varias razones. En primer lugar su análisis no queda completo mediante un simple examen del contenido y la extracción de metadatos, etiquetas y mensajes de texto. Con frecuencia también es necesario entender y explicar su funcionamiento, algo que no está al alcance de cualquiera. Para ello se necesitan sólidas nociones de programación y una experiencia considerable en el manejo de herramientas de desarrollo: editores de código, compiladores, desensambladores, etc. El investigador, sobre todo si está trabajando con una plataforma Windows, también necesitará un entorno seguro –una caja de arena o *sandbox*– para evitar que la ejecución de los archivos sospechosos con fines experimentales pueda dañar su sistema. Este entorno seguro puede consistir en una máquina virtual o una plataforma de hardware aislada.

Normalmente un archivo ejecutable (en DOS/Windows se reconocen por las extensiones EXE y COM) está compuesto por una secuencia de instrucciones en código binario cuyo destino es ser procesadas directamente por la CPU. Dentro de él puede haber información adicional y texto en forma de mensajes de error y comentarios insertados con finalidades explicativas o de documentación. No hay que confundir los archivos ejecutables con el código fuente de un programa. Este último está compuesto por las instrucciones del mismo escritas en un lenguaje de alto nivel que después ha de ser procesado por un compilador. El código ejecutable tampoco es lo mismo que los *scripts* de comandos creados para la *bash* del sistema o en lenguajes interpretados como Python o Perl –lo cual no quiere decir que para este tipo de objetos no haya que seguir las mismas precauciones que en el caso de los binarios, con el objeto de evitar que una ejecución accidental de un código malicioso ponga en peligro el sistema del investigador forense–. La definición de código ejecutable no se limita a los archivos binarios. También comprende las librerías de funciones (con extensión DLL), los controladores de dispositivos y naturalmente el *kernel* o núcleo del sistema.

En un nivel básico, cada arquitectura de hardware dispone de su propio código ejecutable. Sin embargo el formato de los archivos ejecutables depende del sistema operativo. Antes de que un archivo binario pueda ejecutarse el sistema operativo ha de llevar a cabo gran cantidad de tareas preparatorias, lo cual resultaría imposible de no existir un formato interno preciso y rígidamente formalizado para los programas destinados a ser procesados directamente por la CPU. Dicho formato permite que el código sea copiado desde el disco duro a la RAM antes de su ejecución y contiene la información necesaria para que el sistema operativo sepa si el código es compatible o no, si se trata de una aplicación para línea de comando o entorno gráfico, y sobre todo para construir las tablas de importación de funciones existentes en las librerías DLL, en caso de necesitar recursos de código compartido del sistema. Aunque existen diversos formatos de archivo ejecutable, los más utilizados en la actualidad son Windows PE, ELF de Linux y Apple OSX. En cuanto a herramientas de análisis, los especialistas se sirven de IDA Pro, un desensamblador comercializado por la empresa belga DataRescue.

### 3.6.6 Exclusión de archivos conocidos

En el disco duro de un ordenador de sobremesa equipado con Windows 7 y las aplicaciones de productividad habituales (MS-Office, Photoshop, etc.) puede haber fácilmente en torno a los ciento cincuenta o doscientos mil archivos. El examen de todo este material, incluso con herramientas automatizadas, requiere una cantidad considerable de tiempo y ciclos de procesador. ¿Para qué analizar sin embargo elementos como ejecutables del sistema, librerías, plantillas de Office o

archivos de ayuda, que suelen ser iguales en todas las instalaciones de la misma versión de un sistema operativo provisto del software de usuario habitual? Resultaría más práctico excluirlos del análisis y concentrarse en lo que realmente interesa, como documentos de trabajo del usuario, archivos de registro y ejecutables de procedencia extraña. ¿Cómo podemos reducir la carga de trabajo del ordenador y las herramientas de análisis forense excluyendo todos aquellos archivos que no resulten de interés para la investigación?

La respuesta está en servicios de autentificación mediante *hash*. El recurso a una base de datos con las firmas MD5 o SHA de aquellos archivos conocidos que forman parte de sistemas operativos y aplicaciones estándar permite reducir de manera drástica el volumen de material para el análisis. El procedimiento consiste en comprobar los archivos uno por uno para ver si su *hash* coincide con la del archivo correspondiente en la base de datos (por ejemplo el ejecutable de MS-Excel 2003 o la librería devobj.dll en Windows 7). Si fuera así lo eliminaríamos del inventario de objetos del caso, y así consecutivamente con todos los demás que se encuentren en el mismo caso, dejando al final un número de archivos relativamente reducido sobre el cual podemos concentrar nuestro esfuerzo con mayor eficacia.

Para facilitar la labor existen listas de archivos conocidos con sus *hashes* correspondientes elaboradas por agencias públicas como NIST (*National Institute of Standards and Technology*) y Hashkeeper. Estas listas se encuentran en Internet a disposición del usuario. Algunas herramientas comerciales como FTK las importan de modo automático para reducir la carga de trabajo en las investigaciones forenses. El software comercial EnCase de Guidance Forensics también posee una funcionalidad denominada Hash Category, que viene a ser el equivalente a los KFF (*Known File Filter*) de FTK.

## 3.7 DATA CARVING

Volvamos a la llave USB del apartado 3.5. ¿Recuerda cómo habíamos analizado su contenido con TSK, localizando archivos PDF borrados por el usuario? Incluso habíamos conseguido resucitar a uno de ellos? Estuvimos trabajando duro para atrapar nuestro pequeño elemento de evidencia, pero no nos conformamos con eso. TSK nos dice lo que hay dentro de un soporte. También es capaz de encontrar archivos borrados. Pero no nos cuenta nada sobre lo que pudiera haberse guardado en el medio antes de formatearlo. Las llaves USB de 256 MB dejaron de fabricarse hace años. El historial de deslealtad corporativa de este soporte de datos podría prolongarse hasta muy atrás en el tiempo. La mayor parte de los archivos antiguos se habrán perdido al sobreescribir datos nuevos sobre ellos, pero algo podría haber quedado en el espacio no utilizado del medio de

almacenamiento. No lo sabemos, simplemente es una suposición. Las herramientas de “tallado de archivos” o *data carving* nos ayudarán a aclarar la duda.

### 3.7.1 Cuando todo lo demás falla

Las operaciones de *data carving* se basan en un principio simple: llevar a cabo una lectura del soporte, de modo secuencial y a bajo nivel, al margen de las estructuras de datos del sistema de archivos, en busca de firmas características (ver apartado 3.6.1) que nos permitan detectar la presencia de un archivo y extraerlo a un medio de destino después de haber hecho algunas suposiciones razonadas acerca de su tamaño y el número de bloques consecutivos que lo componen. Por lo general una herramienta de *data carving*, cada vez que encuentra un archivo, copia todos los bloques contiguos que vienen a continuación hasta encontrar uno que contenga el próximo número mágico, y entonces repetirá el proceso con el archivo siguiente. Esto supone una limitación cuando los archivos están fragmentados, pero no queda otro remedio porque no disponemos de estructuras de datos correspondientes al sistema de archivos preexistente. Sin embargo en soportes de gran capacidad la posibilidad de fragmentación, a no ser que se trate de archivos extensos como películas, audio o materiales multimedia, tiende a reducirse haciendo más ventajoso el empleo de herramientas de *data carving*. Esto resulta especialmente válido para los discos duros modernos, que con tamaños descomunales en el orden de 1,5 o incluso 2 Terabytes, en la práctica ni siquiera los adictos a las descargas llegan a llenar del todo durante la vida útil del producto.

### 3.7.2 Extracción de archivos

El *data carving* es una funcionalidad integrada en herramientas forenses comerciales como EnCase y FTK. También existen utilidades específicas de código libre como foremost, un programa en línea de comando que viene incluido en la mayor parte de las distribuciones de Linux dedicadas a la seguridad. Para realizar su trabajo foremost se guía por las cabeceras y estructuras de datos de los archivos. No solo es capaz de procesar directamente imágenes forenses adquiridas con dd, sino que además admite el formato de imagen extendido de EnCase.

Para instalar foremost desde los repositorios de Ubuntu:

```
sudo apt-get install foremost
```

También puede descargarlo desde la página del desarrollador, compilarlo e instalarlo del modo expuesto con anterioridad en este mismo capítulo. La ejecución de foremost es simple:

```
foremost imagen0.dd
```

Si en lugar de un *pendrive* de 256 MB la imagen fuera de un disco duro de 500 GB, el investigador deberá tomárselo con calma y disponer de un medio de destino lo suficientemente grande. Por supuesto foremost también funciona sobre soportes de hardware. Al actuar con independencia de las estructuras del sistema de archivos foremost no recupera nombres ni marcas de tiempo, sino que se limita a asignar un número de referencia a cada archivo extraído, junto con la fecha y hora correspondientes al momento en que se lleva a cabo la operación de *data carving*. Los resultados se almacenan en un directorio llamado *output*, el cual contiene a su vez varios subdirectorios, uno por cada tipo de archivo encontrado. También se elabora automáticamente un informe de auditoría.

Mediante su archivo de configuración *foremost.conf*, la herramienta permite especificar el tipo de archivos en los que debe centrar su búsqueda. No es necesario utilizar ningún archivo de configuración en el directorio de trabajo. De no haberlo, el programa se ejecutará con las opciones estándar indicadas en su archivo de configuración por defecto */etc/foremost.conf*. En nuestro caso foremost nos permite comprobar que, efectivamente, el sospechoso tenía varios archivos gráficos guardados en su llave USB antes de formatearla. Un examen de las imágenes y sus metadatos Exif nos dirá si se trata de un hallazgo relevante para la investigación.

## ANÁLISIS FORENSE DE SISTEMAS MICROSOFT WINDOWS

### Capítulo 4

En alguna de sus versiones el sistema operativo MS-Windows se encuentra instalado en nueve de cada diez ordenadores de todo el mundo, junto con aplicaciones de productividad características como MS-Office, procesadores de texto como Word Perfect, el software de retoque fotográfico Adobe Photoshop, lectores de documentos como Acrobat Reader y otros programas por el estilo. Es muy probable que el próximo caso de que el investigador tenga que ocuparse sea el análisis forense de un entorno Windows. En el capítulo sobre soportes se dijo que un sistema de archivos, con su estructura y particularidades, determina los métodos y estrategias aplicables en la búsqueda de elementos de evidencia. Esto resulta particularmente cierto en el caso de los entornos Microsoft, tanto por el uso exclusivo de particiones NTFS y FAT como por el diseño y las características de funcionamiento del sistema.

Existen numerosas herramientas para el estudio forense de sistemas Windows. Algunas de ellas también funcionan sobre otros sistemas operativos como Mac OSX o Linux. Las hay con interfaz gráfica que se manejan de manera intuitiva y práctica a golpe de ratón, y también utilidades en línea de comando. Algunas de las que se van a ver a continuación son de este último tipo. Para utilizarlas será preciso que el lector esté familiarizado con el empleo de la consola. El manejo de estas herramientas no es difícil. Casi todas tienen una sintaxis parecida, consistente en teclear el nombre del programa acompañado del archivo de datos que se quiere procesar. A muchas personas que no conocieron el antiguo MSDOS y se han iniciado en la Informática con interfaces gráficas Windows y Apple la línea de comando les provoca un rechazo instintivo. Sin embargo, sus

incomodidades quedan compensadas por el beneficio de una mayor potencia, versatilidad y precisión en los resultados.

### 4.1 RECOPILANDO INFORMACIÓN VOLÁTIL

En el análisis forense de un entorno Windows resulta imprescindible conocer en primer lugar el estado en que se encuentra cuando el investigador o la persona encargada de hacerse cargo de él llega al lugar de los hechos. El análisis *post mortem* consiste en el estudio forense de las imágenes de soportes de datos practicadas después de haber apagado el ordenador –frecuentemente no de un modo ordenado sino por el método de interrumpir la alimentación eléctrica–. Por el contrario una investigación en vivo comienza con la búsqueda de información relativa a procesos en ejecución, conexiones de red, usuarios conectados, archivos abiertos y todo aquello que por estar en RAM se perdería en caso de apagado o reinicio del equipo, sin posibilidad ninguna de reproducirlo con posterioridad a partir de un duplicado forense del disco duro. Esto es lo que llamamos información volátil, la cual puede contener elementos de evidencia decisivos como por ejemplo los siguientes:

- Fecha y hora del sistema.
- Conexiones de red abiertas.
- Puertos TCP o UDP abiertos.
- Ejecutables a la escucha en puertos TCP o UDP.
- Usuarios conectados al sistema.
- Tabla de enrutamiento interna.
- Procesos en ejecución.
- Archivos abiertos.

#### 4.1.1 Fecha y hora del sistema

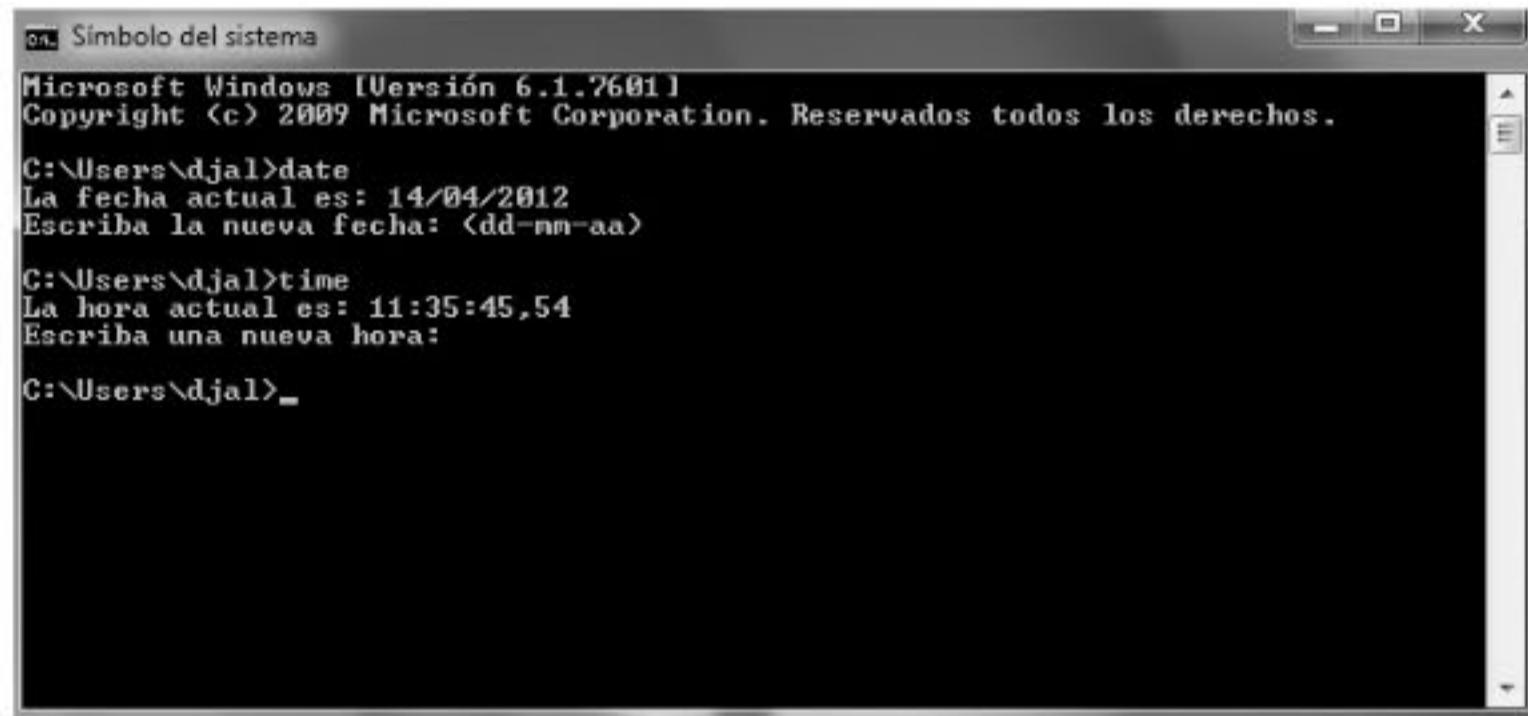
La fecha y la hora indicadas por el reloj del sistema son importantes para establecer cualquier diferencia que pueda existir con respecto a la fecha y hora reales así como posibles variaciones relacionadas con el traslado del equipo de unos usos horarios a otros. Si quiere estar en condiciones de elaborar una línea de tiempo coherente el investigador debe tener un buen reloj de pulsera y llevarlo siempre a la hora, como aún exige el reglamento del personal ferroviario en algunos países.

Para obtener la fecha y hora del sistema basta abrir una consola de texto (**Inicio → Ejecutar**, y allí: cmd.exe) y teclear lo siguiente:

```
C:\Documents and Settings\XYZW>date
```

```
C:\Documents and Settings\XYZW>time
```

Fecha y hora son importantes sobre todo a la hora de relacionar los datos del equipo con los de otras máquinas, suponiendo que la investigación no esté limitada a un solo ordenador.



```
Microsoft Windows [Versión 6.1.7601]
Copyright © 2009 Microsoft Corporation. Reservados todos los derechos.

C:\>date
La fecha actual es: 14/04/2012
Escriba la nueva fecha: <dd-mm-aa>

C:\>time
La hora actual es: 11:35:45,54
Escriba una nueva hora:

C:\>
```

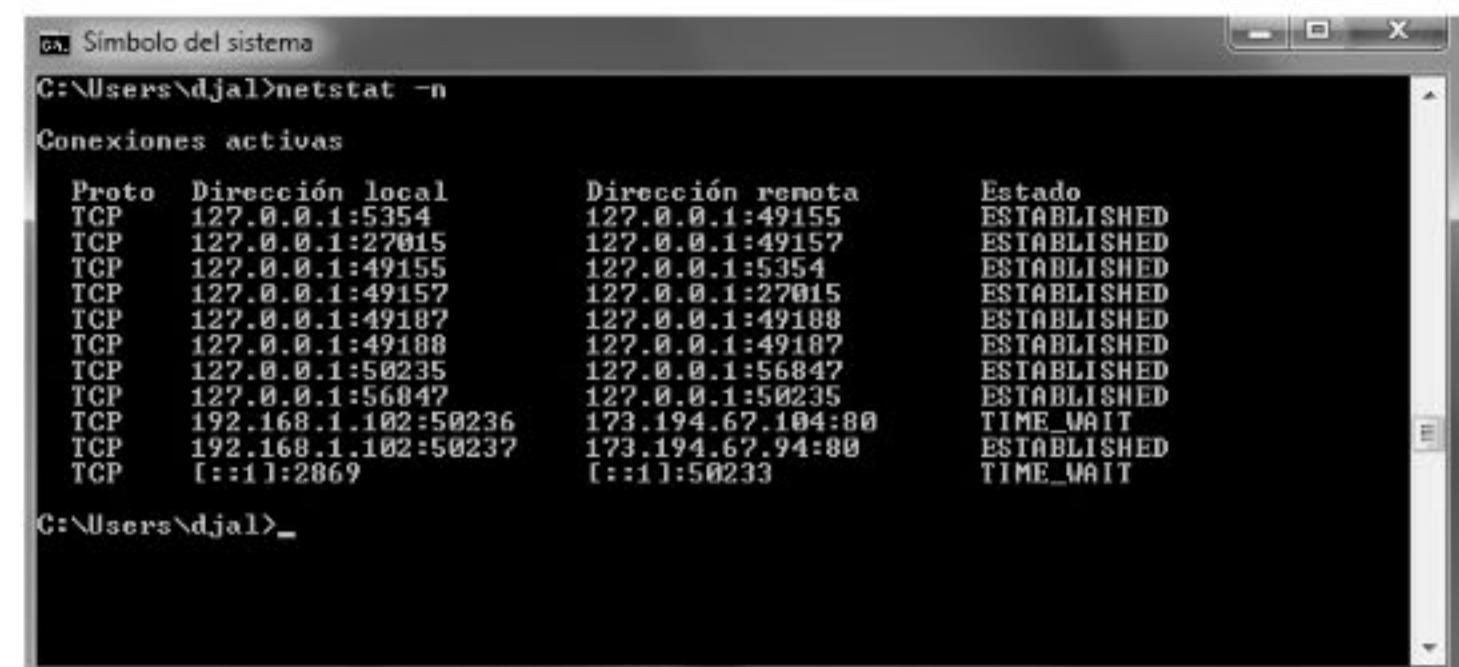
Figura 4.1. Fecha y hora del sistema a través de la consola de texto

#### 4.1.2 Conexiones de red abiertas

Si estamos investigando un servidor y nos disponemos a adquirir su información volátil cabe la posibilidad de que en esos momentos el atacante aún esté conectado a él desde un equipo remoto. Lo podemos comprobar examinando las conexiones de red. Desde la misma consola de texto con la que obtuvimos la fecha y hora del sistema, y que tenemos abierta todavía, tecleamos el comando:

```
C:\Documents and Settings\XYZW>netstat -n
```

El resultado muestra (figura 4.2) todas las conexiones que el sistema mantiene abiertas, indicando direcciones IP locales y remotas, tipo de conexión (TCP/UDP) y estado de la misma: a la escucha (LISTENING), establecida (ESTABLISHED), cerrada (CLOSE\_WAIT y TIME\_WAIT). Por este procedimiento sabremos si puede haber programas conectados con ubicaciones remotas o alguien accediendo al sistema desde una IP que nos parezca sospechosa.



```
Microsoft Windows [Versión 6.1.7601]
Copyright © 2009 Microsoft Corporation. Reservados todos los derechos.

C:\>netstat -n
Conexiones activas

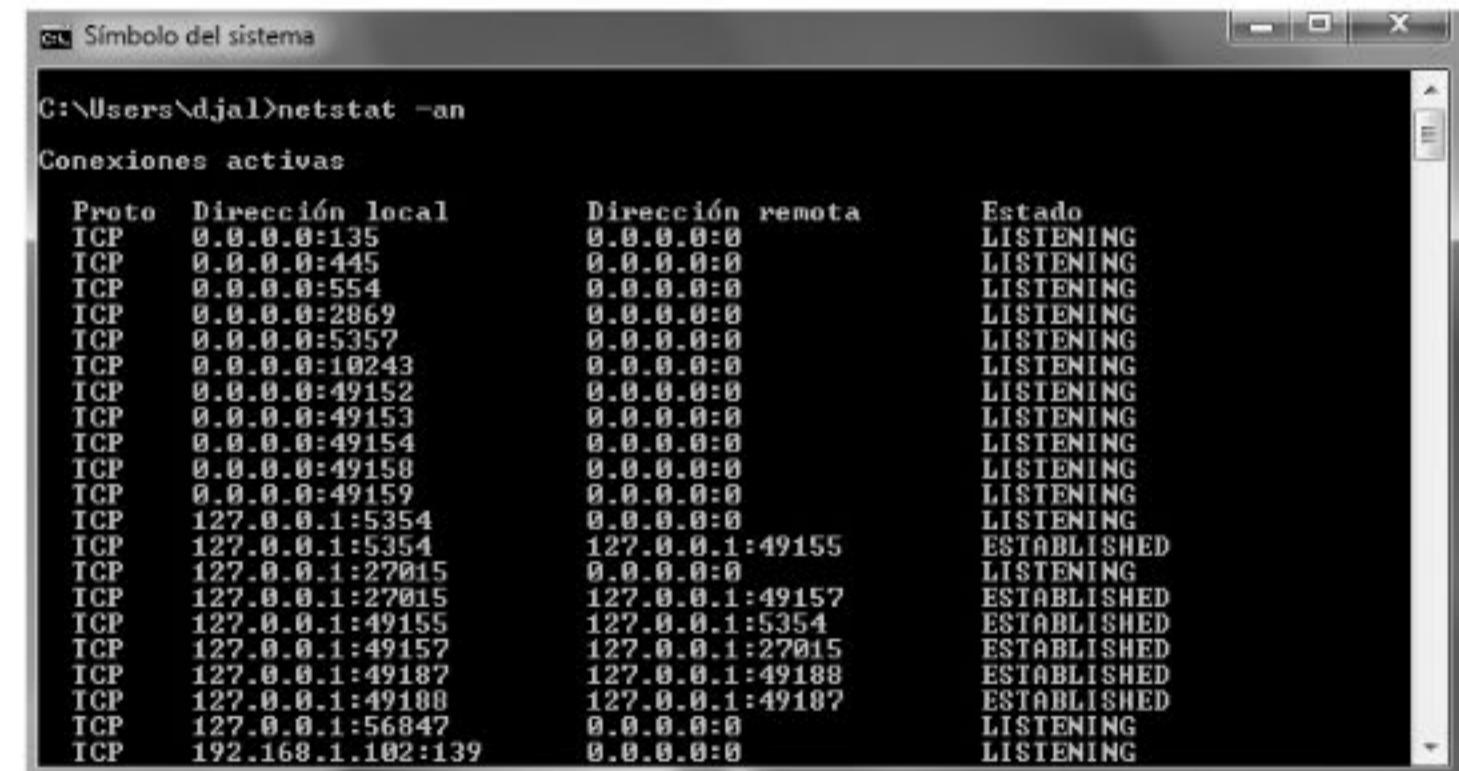
Proto Dirección local           Dirección remota         Estado
TCP  127.0.0.1:5354             127.0.0.1:49155        ESTABLISHED
TCP  127.0.0.1:27015            127.0.0.1:49157        ESTABLISHED
TCP  127.0.0.1:49155            127.0.0.1:5354        ESTABLISHED
TCP  127.0.0.1:49157            127.0.0.1:27015        ESTABLISHED
TCP  127.0.0.1:49187            127.0.0.1:49188        ESTABLISHED
TCP  127.0.0.1:49188            127.0.0.1:49187        ESTABLISHED
TCP  127.0.0.1:50235            127.0.0.1:56847        ESTABLISHED
TCP  127.0.0.1:56847            127.0.0.1:50235        ESTABLISHED
TCP  192.168.1.102:50236         173.194.67.104:80      TIME_WAIT
TCP  192.168.1.102:50237         173.194.67.94:80      ESTABLISHED
TCP  [::1]:2069                [::1]:50233          TIME_WAIT

C:\>
```

Figura 4.2. Listado básico de conexiones mediante netstat

#### 4.1.3 Puertos TCP y UDP abiertos

Si examinamos el listado de *netcat* de la figura 4.3 notaremos dos cosas. En primer lugar la presencia de unos números que figuran asignados mediante dos puntos a direcciones IP. Son los puertos. Segundo, las indicaciones que figuran bajo la columna de Estado en el extremo derecho de la tabla. LISTENING (ESCUCHANDO) significa que el puerto está abierto.



```
Microsoft Windows [Versión 6.1.7601]
Copyright © 2009 Microsoft Corporation. Reservados todos los derechos.

C:\>netstat -an
Conexiones activas

Proto Dirección local           Dirección remota         Estado
TCP  0.0.0.0:135              0.0.0.0:0          LISTENING
TCP  0.0.0.0:445              0.0.0.0:0          LISTENING
TCP  0.0.0.0:554              0.0.0.0:0          LISTENING
TCP  0.0.0.0:2869             0.0.0.0:0          LISTENING
TCP  0.0.0.0:5357             0.0.0.0:0          LISTENING
TCP  0.0.0.0:10243            0.0.0.0:0          LISTENING
TCP  0.0.0.0:49152            0.0.0.0:0          LISTENING
TCP  0.0.0.0:49153            0.0.0.0:0          LISTENING
TCP  0.0.0.0:49154            0.0.0.0:0          LISTENING
TCP  0.0.0.0:49158            0.0.0.0:0          LISTENING
TCP  0.0.0.0:49159            0.0.0.0:0          LISTENING
TCP  127.0.0.1:5354            0.0.0.0:0          LISTENING
TCP  127.0.0.1:5354            127.0.0.1:49155        ESTABLISHED
TCP  127.0.0.1:27015            0.0.0.0:0          LISTENING
TCP  127.0.0.1:27015            127.0.0.1:49157        ESTABLISHED
TCP  127.0.0.1:49155            127.0.0.1:5354        ESTABLISHED
TCP  127.0.0.1:49157            127.0.0.1:27015        ESTABLISHED
TCP  127.0.0.1:49187            127.0.0.1:49188        ESTABLISHED
TCP  127.0.0.1:49188            127.0.0.1:49187        ESTABLISHED
TCP  127.0.0.1:56847            0.0.0.0:0          LISTENING
TCP  192.168.1.102:139          0.0.0.0:0          LISTENING
```

Figura 4.3. Puertos abiertos

Todo puerto abierto supone una puerta trasera en el equipo local. En el transcurso de su funcionamiento normal Windows deja abiertos un número de

puertos de uso legítimo –por ejemplo el 139 y el 445, necesarios para el protocolo NetBIOS y otros servicios de red–. Con la ayuda de netstat, sin embargo, resulta imposible diferenciar el trigo de la mala hierba. Tampoco sabemos qué proceso hay detrás de cada puerto.

#### 4.1.4 Ejecutables conectados a puertos TCP y UDP

Para averiguar si alguno de esos puertos ha sido habilitado por un programa parásito con la intención de establecer contacto con una aplicación remota es preciso conocer los procesos que se están ejecutando en el sistema y tienen puertos abiertos. La herramienta FPort, distribuida libremente en [www.foundstone.com](http://www.foundstone.com), nos ayudará a averiguarlo. Se trata de un programa en línea de comando que relaciona la información de netstat con archivos ejecutables. En caso de que alguno de los procesos que el sistema está ejecutando tenga asignado un puerto, FPort nos lo dirá:

```
C:\Documents and Settings\XYZW> fport
```

En el ejemplo de la figura 4.4 tenemos una situación normal con los procesos habituales del sistema y un navegador Internet Explorer iniciado por el usuario:

```

C:\Trabajo>fport
FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

Pid  Process      Port Proto Path
936  System       -> 135  TCP
4   System       -> 139  TCP
4   System       -> 445  TCP
544  System       -> 1025 TCP
476  iexplore    -> 1222 TCP  C:\Archivos de programa\Internet Explorer\i
explore.exe
476  iexplore    -> 1223 TCP  C:\Archivos de programa\Internet Explorer\i
explore.exe
476  iexplore    -> 1224 TCP  C:\Archivos de programa\Internet Explorer\i
explore.exe
0   System       -> 123   UDP
476  iexplore    -> 123   UDP  C:\Archivos de programa\Internet Explorer\i
explore.exe
544  System       -> 137   UDP
0   System       -> 138   UDP
936  System       -> 445   UDP
4   System       -> 500   UDP
4   System       -> 1026  UDP
476  iexplore    -> 1131  UDP  C:\Archivos de programa\Internet Explorer\i
explore.exe
0   System       -> 1221  UDP
0   System       -> 1900  UDP
476  iexplore    -> 4500  UDP  C:\Archivos de programa\Internet Explorer\i
explore.exe

```

Figura 4.4. Salida de FPort

¿A qué viene tanto interés por puertos y procesos? La respuesta tiene que ver con la manera en que se producen las intrusiones en los sistemas. El atacante ejecuta un *exploit* que le permite aprovecharse de un fallo en el sistema operativo, o bien convence al usuario para visitar una página web con código malicioso mediante mensajes de *spam*, llamadas telefónicas o a través de cualquier otra maniobra de ingeniería social. El objetivo de estas acciones consiste en instalar en el ordenador de su víctima un programa que al ejecutarse establezca una conexión con la máquina remota del atacante. De este modo queda despejada la vía de acceso para dominar el ordenador de la víctima. A partir de ese momento resultará posible robar información personal del usuario o agregar su ordenador a una *botnet* cuyo objetivo consiste en realizar envíos masivos de correo no solicitados, lanzar ataques de denegación de servicio contra sitios web o llevar a cabo otras acciones ilícitas. El agresor también podrá habilitar parte del disco duro del ordenador invadido para convertirlo en un repositorio de *warez* o pornografía infantil.

Este es el modo operativo habitual de troyanos, *rootkits* y otros elementos de software malicioso, que a menudo escapan a los antivirus y las herramientas de detección de software espía. La única manera de combatirlo con eficacia es mediante el examen detallado y metódico de las conexiones abiertas así como de los ejecutables que las establecen y los puertos abiertos por aquellos para comunicarse con el exterior.

#### 4.1.5 Usuarios conectados al sistema

Así mismo interesa conocer si en el momento de llegar al lugar de los hechos hay alguien conectado al sistema, accediendo a recursos compartidos o realizando tareas de cualquier otro tipo. Para ello utilizamos PsLoggedOn, perteneciente a la *suite* de herramientas PsTools de la empresa Sysinternals, fundada por Mark Russinovich. Las herramientas PsTools proporcionan información de procesos MS-Windows. Estas herramientas estuvieron alojadas en la página web de Mark Russinovich, su creador, hasta que Sysinternals fue adquirida por Microsoft. No obstante es el propio Mark Russinovich quien se sigue ocupando de su mantenimiento, ahora por cuenta de su nuevo patrono. Las herramientas PsTools continúan estando a disposición del público y se pueden descargar gratuitamente desde el sitio Technet de Microsoft.

En este caso abrimos la consola de texto y tecleamos:

```
C:\Documents and Settings\XYZW> psloggedon
```

Si hay algún usuario conectado al sistema aparecerán tanto las direcciones local y remota como los puertos respectivos. Esto nos permitirá conocer la IP del

atacante. Un puerto 445 (NetBIOS) asociado a la dirección local significa que existe acceso a recursos compartidos.

#### 4.1.6 Tabla de enrutamiento interna

Examinando la tabla de enrutamiento se puede saber si el atacante desvía el tráfico de red para sortear cortafuegos o sistemas de detección de intrusos (IDS). Otro de sus objetivos puede consistir en desviar el tráfico hacia un punto en el que resulte posible monitorizarlo mediante un escáner de red. Para ver la tabla de enrutamiento utilizamos el comando netstat con los parámetros -r n (figura 4.5).

```

Símbolo del sistema
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\djal>netstat -rn
=====
Lista de interfaces
11...00 25 22 93 de 52 .....Realtek PCIe FE Family Controller
1.....Software Loopback Interface 1
12...00 00 00 00 00 00 e0 Adaptador ISATAP de Microsoft
13...00 00 00 00 00 00 e0 Teredo Tunneling Pseudo-Interface
=====

IPv4 Tabla de enrutamiento
Rutas activas:
Destino de red    Máscara de red    Puerta de enlace    Interfaz    Métrica
0.0.0.0          0.0.0.0          192.168.1.102      20
127.0.0.0        255.0.0.0        En vínculo          127.0.0.1    306
127.0.0.1        255.255.255.255  En vínculo          127.0.0.1    306
127.255.255.255 255.255.255.255  En vínculo          127.0.0.1    306
192.168.1.0      255.255.255.0    En vínculo          192.168.1.102 276
192.168.1.102   255.255.255.255  En vínculo          192.168.1.102 276
192.168.1.255   255.255.255.255  En vínculo          192.168.1.102 276
224.0.0.0        240.0.0.0        En vínculo          127.0.0.1    306
224.0.0.0        240.0.0.0        En vínculo          192.168.1.102 276
255.255.255.255 255.255.255.255  En vínculo          127.0.0.1    306
255.255.255.255 255.255.255.255  En vínculo          192.168.1.102 276
=====
Rutas persistentes:
Ninguno
=====

IPv6 Tabla de enrutamiento
Rutas activas:
Cuando destino de red métrica    Puerta de enlace
13 58 ::/0                      En vínculo
1 306 ::1/128                    En vínculo
13 58 2001:::/32                 En vínculo
13 306 2001::0:5ef5:79fb:344e:3c58:aaab:1fd3/128  En vínculo
11 276 fe80::/64                 En vínculo
13 306 fe80::/64                 En vínculo
11 276 fe80::92f:daf6:d3ad:66fc/128  En vínculo
13 306 fe80::344e:3c58:aaab:1fd3/128  En vínculo
1 306 ff00::/8                   En vínculo
13 306 ff00::/8                   En vínculo
11 276 ff00::/8                   En vínculo
=====
Rutas persistentes:
Ninguno
=====

C:\Users\djal>

```

Figura 4.5. Tabla de enrutamiento incluyendo parámetros Ipv6

#### 4.1.7 Procesos en ejecución

Necesitamos saber si además de los procesos normales del sistema y las aplicaciones de usuario puede haber activo algún proceso dejado por el atacante y revelador de algún tipo de actividad en el ordenador que pudiera considerarse sospechosa. Para ello podríamos servirnos del gestor de tareas de Windows, llamándolo con la combinación de teclas **Control+Alt+Supr**, pero no es una buena idea. Estamos recopilando la información volátil de un sistema como paso previo a una posible adquisición forense del mismo en su totalidad. Lo que nos interesa es la actividad del sospechoso, no la originada por nuestros propios comandos. Las aplicaciones gráficas consumen gran cantidad de recursos, y lo que nosotros queremos es que el impacto de las acciones llevadas a cabo por el investigador quede reducido al mínimo. De manera que en este caso resulta más práctico utilizar Pslist:

```

C:\Documents and Settings\XYZW> pslist
=====
C:\Windows\system32\cmd.exe
D:\IRsys\internals>pslist
pslist v1.28 - Sysinternals PsList
Copyright © 2000-2004 Mark Russinovich
Sysinternals
Process information for LEUZEN:
Name          PID Pri Thd Hnd Pri CPU Time Elapsed Time
Idle          0 0 2 0 0 1:05:41.093 0:00:00.000
System         4 0 104 558 48 0:00:18.515 0:36:41.437
smss          272 11 2 38 256 0:00:00.031 0:36:41.406
csrss         412 11 5 525 1448 0:00:00.250 0:00:38.640
wininit       468 13 3 79 848 0:00:00.046 0:36:38.453
csrss         475 13 48 241 2048 0:00:00.145 0:36:38.453
services      545 9 13 222 5008 0:00:00.765 0:36:38.374
lsass         549 9 8 755 2236 0:00:00.000 0:36:38.342
lsass         549 8 18 155 2268 0:00:00.180 0:36:38.342
lsass         549 8 18 264 2232 0:00:00.090 0:36:38.342
lsass         549 8 18 313 1636 0:00:00.291 0:36:38.248
lsass         789 8 11 364 2232 0:00:00.090 0:36:38.248
svchost       784 8 9 313 2992 0:00:00.312 0:36:33.171
svchost       849 8 23 587 15200 0:00:00.421 0:36:33.148
svchost       924 8 29 525 5212 0:00:00.921 0:36:33.051
svchost       964 8 39 1241 17452 0:00:00.931 0:36:33.015
svchost       1116 8 17 498 6548 0:00:00.296 0:36:32.593
svchost       1216 8 15 295 8932 0:00:00.328 0:36:32.499
spoolsv      1344 8 13 228 4656 0:00:00.093 0:36:32.249
shed          1380 8 8 105 2496 0:00:00.283 0:36:32.197
svchost       1436 8 19 317 9816 0:00:00.750 0:36:31.562
svguard       1500 8 25 223 136396 0:00:31.375 0:36:31.484
AppleMobileDeviceService 1536 8 19 201 2284 0:00:00.079 0:36:31.4
=====
taskhost      1616 8 8 203 7112 0:00:00.140 0:36:31.342
DNSResponder  1649 4 4 108 1412 0:00:00.187 0:36:31.296
dum          1736 13 6 105 47464 0:00:17.593 0:36:31.294
svchost       1760 8 28 402 5776 0:00:00.432 0:36:31.293
explorer      1776 8 31 1116 30920 0:00:00.109 0:36:31.293
svchost       1900 8 6 95 1044 0:00:00.080 0:36:31.046
HelloDUDapl  1672 8 9 239 7304 0:00:00.109 0:36:30.320
iglxtray     1640 8 2 1132 0:00:00.815 0:36:30.312
hklm         1776 8 2 1108 0:00:00.815 0:36:30.312
iglxpers     1024 8 2 98 1056 0:00:00.891 0:36:30.276
avgnat       1084 8 16 145 5128 0:00:00.265 0:36:30.265
iglxexec     2800 8 4 76 1548 0:00:00.250 0:36:30.046
ARM Updates Notifier 2276 0 5 278 5284 0:00:00.500 0:36:29.592
ilunesHelper  2290 0 18 224 2980 0:00:00.831 0:36:29.577
OfferBox      2364 0 6 219 76096 0:00:00.203 0:36:29.350
OfferboxHTTPProxy 2544 0 7 150 3952 0:00:00.531 0:36:28.137
avshdav      3072 0 3 56 986 0:00:00.000 0:36:17.604
conhost      3180 0 3 38 463 0:00:00.000 0:36:17.560
SearchIndexer 3196 0 12 609 17654 0:00:00.373 0:36:17.529
iPodService   3196 0 12 104 2632 0:00:00.015 0:36:16.713
wmpavdsk    3440 0 13 436 7636 0:00:00.015 0:36:16.541
svchost       3736 0 9 354 9148 0:00:00.020 0:36:14.589
usancnt      3900 0 6 306 1500 0:00:00.070 0:33:15.803
UIWORD       3532 0 6 334 29422 0:01:00.404 0:32:50.272
firefox      3700 0 38 618 99120 0:04:00.937 0:22:45.542
plugincontainer 1140 0 9 208 17520 0:00:00.460 0:22:20.300
audiodg      3932 0 10 358 31440 0:00:00.359 0:07:01.589
helix        2316 0 11 261 33292 0:00:00.140 0:06:56.069
cmd          2496 0 1 28 1672 0:00:00.000 0:04:31.575
conhost      3436 0 2 53 246 0:00:00.093 0:04:31.564
PSLIST       1060 13 1 147 1972 0:00:00.093 0:00:00.306
=====

D:\IRsys\internals>

```

Figura 4.6. Pslist

Plist muestra en pantalla un listado de procesos acompañados de sus números PID correspondientes junto con información sobre el espacio ocupado en memoria y el tiempo que llevan en funcionamiento desde el arranque del sistema. Esto permite poner los procesos sospechosos en un contexto cronológico adecuado diferenciándolos de las aplicaciones del usuario y los procesos normales del sistema. Estos últimos serán aquellos que se pusieron en marcha al iniciarse el sistema y por consiguiente los que llevan más tiempo funcionando.

Si examina detenidamente la figura 4.6 podrá observar que todos los procesos y servicios normales del sistema (System, wininit, svchost, etc.) llevan aproximadamente el mismo tiempo ejecutándose. Que haya algunos de ellos repetidos no quiere decir que sean falsos. A menudo el sistema necesita iniciar para sus propios fines varias instancias de un mismo proceso. Sin embargo, si encontrara un proceso con el mismo nombre que cualquier otro de los que se ejecutan durante el arranque del sistema, pero que al parecer se hubiera iniciado en un momento muy posterior, habría motivos para sospechar de él. Los atacantes acostumbran a bautizar los ejecutables de sus aplicaciones con nombres iguales o parecidos a los de los procesos legítimos del sistema con el objeto de hacer que pasen desapercibidos. Plist es una aplicación que le será de gran ayuda a la hora de diferenciar el trigo de la paja.

Por razones de seguridad se recomienda ejecutar Plist no desde el propio disco duro del sistema que se está investigando, sino desde una llave USB o el CD-ROM en el cual el investigador lleva sus herramientas forenses. Así mismo sería conveniente utilizar una versión de Plist compilada estáticamente, que lleve consigo todo el código necesario para su ejecución y no necesite recurrir a las librerías del sistema. De este modo no solo se reduce el impacto potencial sobre los datos que han de ser adquiridos, sino que el investigador también se evitará inconvenientes debidos a la posible presencia de *rootkits* o troyanos en el sistema.

#### 4.1.8 Archivos abiertos

¿Qué más puede haber estado haciendo el intruso? ¿Y si alguno de los archivos del equipo estuviera abierto desde una ubicación remota. Para comprobar si hay archivos abiertos ejecutamos otro comando de la suite PsTools, psfile:

```
C:\Documents and Settings\XYZW> psfile
```

### 4.2 ANÁLISIS FORENSE DE LA RAM

Tradicionalmente la investigación forense se ha centrado en la adquisición de soportes de datos y el análisis *post mortem*, dejando de lado los procedimientos

de captura de memoria RAM. Principalmente ello se debía a la falta de métodos, técnicas y herramientas adecuadas. Sin embargo, durante los últimos años la captura de memoria RAM y los procesos contenidos en la misma ha cobrado una importancia crucial en las operaciones de adquisición de elementos de evidencia volátil. El contenido de la RAM puede resultar decisivo en una investigación, ya que además de información relativa a conexiones y procesos –anteriormente recopilada con las herramientas administrativas del sistema y las PsTools– también incluye las líneas de comando tecleadas por el atacante, imágenes completas de los ejecutables y contraseñas en texto claro. En la memoria RAM incluso podrían haber quedado rastros de procesos antiguos ejecutados por el atacante.

Las características de los diferentes sistemas operativos, en combinación con las particularidades técnicas específicas de la gestión de memoria, convierten la adquisición de la RAM en una tarea extremadamente compleja. Para superar estas dificultades se han desarrollado diversas estrategias.

#### 4.2.1 Captura de RAM completa con dd

Una versión especial del comando de Unix/Linux dd adaptada para Windows por George Garner no solo permite adquirir particiones y volúmenes, sino también hacer volcados completos de la memoria del sistema. El acceso a la RAM se efectúa a través de un objeto del sistema operativo denominado /PhysicalMemory, llevándose a cabo a través de netstat la transferencia de su contenido a un archivo ubicado en un disco duro externo u otro ordenador de la red. La letra de unidad dependerá de la configuración del sistema y de la forma en que hayamos conectado el medio de destino:

```
dd.exe if=\\physicalmemory of=e:\evidencia\memory_dump.dd bs=4096
```

El único inconveniente es que este procedimiento no sirve para las versiones posteriores a Microsoft Windows Server 2003 con Service Pack 1, puesto que en ellas el objeto PhysicalMemory, debido a consideraciones de seguridad, ha dejado de ser accesible desde cualquier utilidad que se esté ejecutando en modo de usuario. Como alternativa a dd se puede emplear una herramienta denominada Win32dd, distribuida por la empresa de consultoría y seguridad informática Moonsols (<http://www.moonsols.com>), que accede a la memoria RAM con la CPU funcionando en modo *kernel*, lo cual hace posible eludir las restricciones de seguridad impuestas por Microsoft. De este modo se pueden realizar volcados de memoria prácticamente para cualquier versión del sistema operativo MS-Windows.

## 4.2.2 Volcado de RAM

La técnica del volcado se basa en provocar un fallo catastrófico del sistema para que este inicie un *dumping* de la RAM, que posteriormente el programador analiza en busca de las anomalías responsables del mal funcionamiento de las aplicaciones o del sistema. En este caso lo que se busca no son fallos de programación sino indicios dejados por el atacante. El volcado de memoria suele ser la única opción en situaciones complicadas, en las que por ejemplo no se dispone de acceso físico al equipo que se quiere investigar, existen defectos de configuración del hardware o se trata de máquinas sin teclado, caso habitual en las salas de servidores.

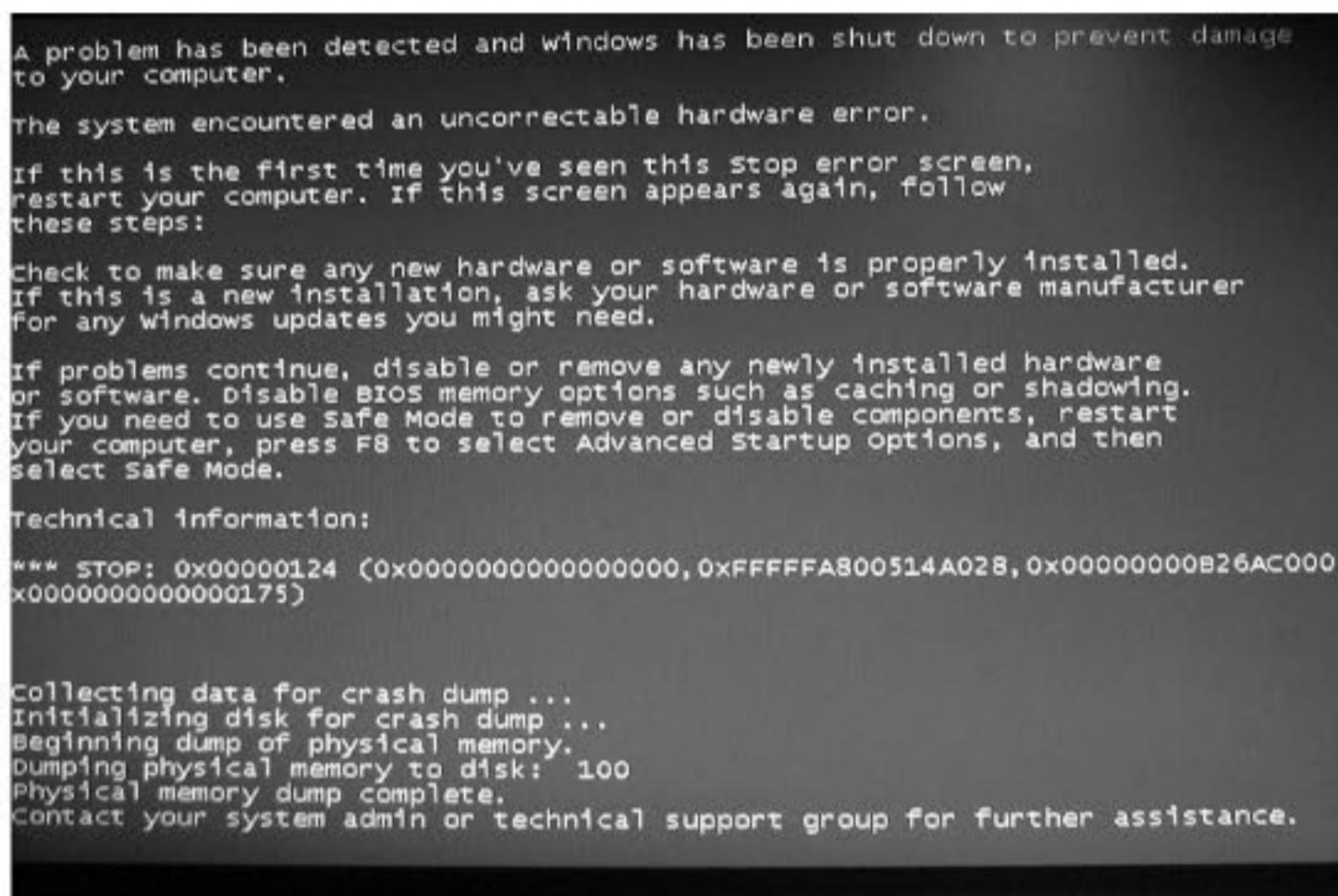


Figura 4.7. Volcado de memoria RAM mediante provocación de un fallo catastrófico

Existen diferentes modos de causar un volcado de RAM. Microsoft recomienda NotMyFault, una herramienta que funciona en combinación con las utilidades Sysinternals –otra aportación de Mark Russinovich además de las PsTools, de las cuales se diferencian por funcionar directamente a través del interfaz gráfico de Windows y no de la consola de texto–. Con NotMyfault se puede generar un BSOD (*Blue Screen Of Death*), el tristemente célebre pantallazo azul de Windows. Otra opción consiste en el empleo de SystemDump, utilidad creada por Dmitry Vostokov. Para saber más sobre SystemDump y volcados RAM en la página web del autor: <http://www.dumpanalysis.org>.

## 4.3 ADQUISICIÓN DE SOPORTES

La adquisición de volúmenes Windows se lleva a cabo por cualquiera de los métodos descritos en el capítulo anterior. No debe olvidarse que en las versiones recientes de Windows (Vista, 7), a la hora de instalar el sistema operativo se habilita automáticamente una partición adicional, reservada para el arranque del sistema y otras operaciones auxiliares. En ordenadores portátiles también es frecuente hallar particiones supplementarias para guardar los datos que permiten reactivar el sistema desde estados de hibernación o ahorro de energía. Habrá que tener todo esto en cuenta a la hora de realizar copias forenses, o de lo contrario podrían perderse datos importantes para la investigación.

Interesa elegir un formato de copia que facilite las tareas posteriores de análisis, asegure la integridad de las pruebas y permita demostrar de manera eficaz el mantenimiento de la cadena de custodia. Los más utilizados son EnCase e imágenes dd. Para adquirir un sistema Windows existen diferentes procedimientos. En los apartados siguientes se consideran tres posibilidades: copia con EnCase, adquisición a través de FTK Imager y captura mediante procedimientos alternativos.

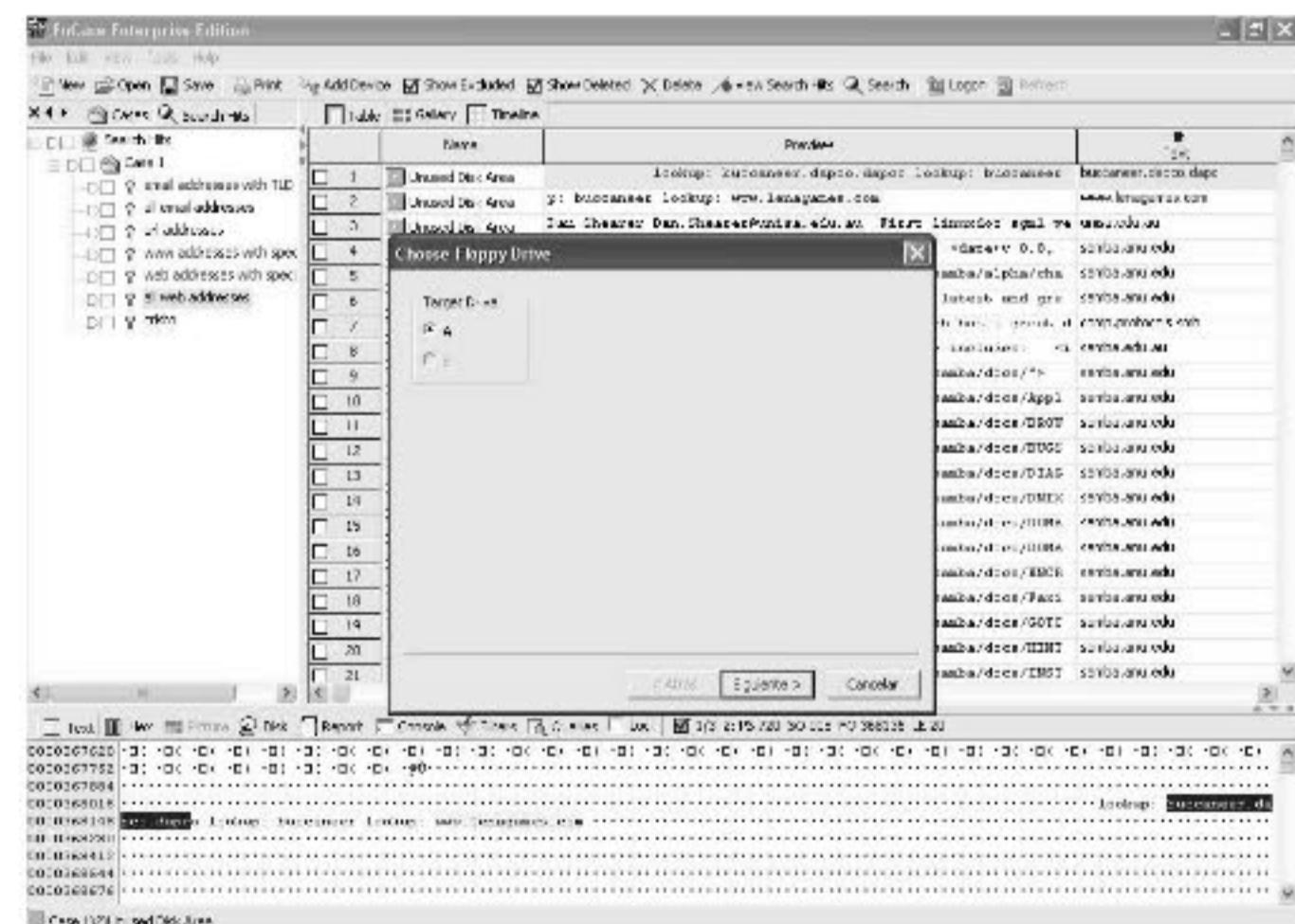


Figura 4.8. Realización de un disquete de arranque con EnCase

### 4.3.1 Adquisición con EnCase

EnCase es un estándar apoyado por administraciones de justicia y departamentos de policía de todo el mundo. Utiliza un formato propio basado en el estándar de imágenes ASR Data Expert Witness. La información adquirida se guarda en archivos con la extensión EXX (donde EXX = E01, E02... etc.). Estos archivos, además de la imagen forense, incluyen información administrativa del caso: nombre del investigador, fecha y hora de adquisición, observaciones y códigos de redundancia CRC. Mediante la herramienta Wipe Drive EnCase también permite higienizar un disco duro o cualquier otro soporte, es decir, realizar un borrado seguro de todos los datos pertenecientes a casos anteriores o a otros usos dados previamente al dispositivo de almacenamiento que no fueran de interés para la investigación en curso o pudieran contaminar los elementos de evidencia adquiridos. El procedimiento habitual consiste en extraer el disco duro del ordenador sospechoso y conectarlo a la estación de trabajo forense a través de un bloqueador de escritura.

La adquisición también se puede llevar a cabo desde un disquete de arranque DOS creado desde la misma interfaz gráfica de EnCase (figura 4.8). En este caso, y una vez creado el disquete, el investigador deberá arrancar desde aquel el ordenador sospechoso después de haber conectado la unidad de destino a uno de los interfaces ATA o SATA disponibles en la placa madre. No olvide llevar un juego de cables y configurar el medio de destino como master o slave mediante los *jumpers* del disco, si se trata de un interfaz ATA. Después de arrancar el ordenador en modo DOS con el disquete el usuario dispone de una opción activada por software para bloquear soportes impidiendo el acceso en modo escritura a la unidad de origen. Cuando se trata de adquirir un disco duro, deberá bloquear este y dejar desbloqueado el medio de destino. El resto del procedimiento se encuentra asistido mediante indicaciones facilitadas desde la espartana pero diáfana interfaz de usuario: seleccionar modo **Adquirir**, localizar el directorio del medio de destino en el que se van a guardar los archivos y el nombre asignado a los mismos. Finalmente, pulsar **ENTER**.

La partición de destino deberá estar formateada con FAT16 o 32 –únicos sistemas de archivos que puede leer el sistema operativo MSDOS–. Asegúrese de disponer de espacio suficiente. El formato EnCase añade 2 Kilobytes al tamaño de la imagen original, correspondientes a información administrativa del caso. Eso quiere decir que los archivos de destino van a ser algo más grandes que los datos adquiridos. Si el investigador utiliza para almacenar la evidencia un disco del mismo tamaño que el medio de origen, la adquisición fracasará a menos que se active la compresión de datos. Finalmente hay que introducir un número para el

caso y el nombre del Investigador. También puede añadirse una descripción del caso. Si la BIOS del sistema indica hora y/o fecha incorrectas, el investigador deberá corregirlas. Otros datos de interés son el número de serie del disco adquirido y los comentarios que el investigador estime oportuno incluir.

¿Es necesaria la compresión? Depende de cuál sea la prioridad: el tiempo (la compresión requiere ciclos de procesador) o el espacio (en caso de que se disponga de un disco de tamaño igual o menor que el medio de origen). Resulta aconsejable calcular el *hash* MD5 durante el proceso de copia, ya que de ese modo podrá demostrarse fácilmente que la imagen del medio no sufrió alteraciones de ningún tipo durante las tareas de análisis. Finalmente el investigador debe considerar la posibilidad de establecer una contraseña de acceso, en caso de que la imagen contenga información confidencial o se deseé limitar el acceso a los datos al personal encargado de la investigación. Una vez seleccionado el tamaño de los archivos de destino (por defecto: 640 MB) puede comenzar el proceso de copia.

### 4.3.2 Adquisición con FTK Imager

AccessData pone a disposición del usuario una herramienta gráfica para Windows llamada FTK Imager (<http://accessdata.com/support/adownloads>) con la cual se pueden imágenes de dispositivos de almacenamiento para fines forenses. FTK Imager, además de posibilitar la adquisición en una variedad de formatos (EnCase, RAW, SMART), permite acceder a las imágenes, extraer de ellas elementos de evidencia y recuperar archivos borrados. Con FTK Imager también se pueden convertir las imágenes de unos formatos a otros. Por su manejo fácil e intuitivo se puede decir que se trata de una aplicación diseñada para investigadores que no tengan mucha experiencia con la línea de comando.

FTK Imager permite adquirir todo aquello que pueda conectarse a la estación de trabajo del analista forense, incluyendo dispositivos USB o Firewire. Sin embargo hay que tener presentes las consideraciones expuestas en capítulos anteriores sobre montaje automático de volúmenes. El empleo de bloqueadores no solo es aconsejable sino en la mayor parte de las situaciones también obligado. Si lo que se quiere adquirir son volúmenes alojados en discos duros externos, *pendrives* o cualquier otro tipo de soporte conectado al ordenador mediante un interfaz USB, resulta aconsejable desactivar en el registro de Windows la funcionalidad de acceso USB en modo escritura. No obstante debe tener en cuenta que toda manipulación del registro puede alterar el funcionamiento del sistema e incluso dejarlo inutilizado. En esta página web hallará información útil acerca de cómo bloquear el acceso a memorias *flash* y dispositivos USB: <http://norfipc.com/virus/flash-usb-solo-lectura.html>.

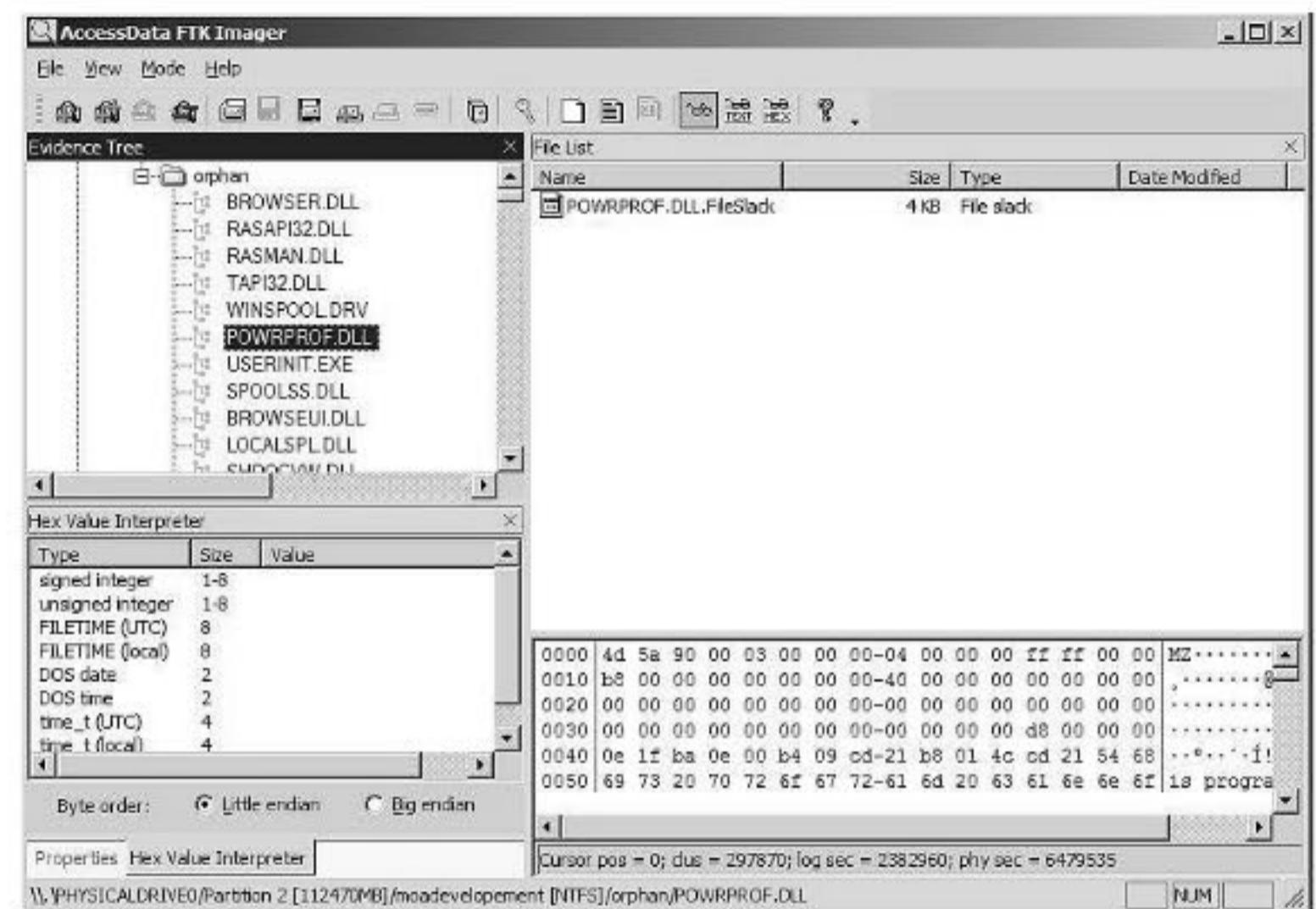


Figura 4.9. FTK Imager

### 4.3.3 Otros métodos

El investigador también puede utilizar dd en sus respectivas versiones Linux o Windows para adquirir discos duros o cualquier otro tipo de soportes de datos. Para una captura *in situ*, sin extraer el disco duro, deberá iniciar el ordenador sospechoso con un CD autoarrancable de Linux y volcar la imagen a un medio de destino. Imagine la siguiente situación: se ha producido un incidente de seguridad y el investigador recibe el aviso de adquirir un disco duro de modo inmediato. La urgencia del caso le impide desplazarse a su laboratorio en busca de herramientas forenses, obligándole a improvisar sobre la marcha. De camino al lugar de los hechos compra en una tienda de informática un disco duro con capacidad suficiente para almacenar los datos de su adquisición forense. Después utiliza uno de los ordenadores de la empresa para descargar una imagen del CD autoarrancable Helix (<http://mirrors.cmich.edu/helix/>). Acto seguido inicia el ordenador sospechoso desde el CD. Una vez cargado el sistema operativo, ejecuta la aplicación Adepto (en **Applications → Forensics & IR → Adepto**) y realiza la imagen del disco duro en alguno de los formatos soportados por el programa (dd o AFF).

Se recomienda emplear bloqueadores de escritura siempre que sea posible, o al menos un software que no monte las particiones con *journaling* en modo escritura. Knoppix puede ser una excelente distribución Linux para finalidades de aprendizaje o reparación de sistemas, pero como herramienta forense no resulta aconsejable ya que monta por defecto las particiones con *journaling* en modo escritura. SystemRescueCD en cambio no.

## 4.4 ANÁLISIS POST MORTEM

Como ya se ha visto en el apartado 3.5, The Sleuth Kit es una herramienta ideal para el estudio de volúmenes y particiones de los tipos más variados. Windows no constituye una excepción. Si estamos utilizando una estación de trabajo con Linux podemos montar la imagen de un soporte de datos adquirido con dd y examinarla a fondo en busca de documentos, archivos borrados y otros elementos de evidencia. Si nuestra plataforma forense es del tipo Windows también podemos trabajar con TSK. Para ello deberemos haber instalado previamente CygWin, un interfaz diseñado para simular entornos Unix dentro de un sistema operativo Microsoft. En la práctica, la forma de proceder viene dada por las herramientas disponibles –sin perjuicio de que para soportes con características especiales haya que buscar otros recursos–. En las páginas que siguen pasaremos revista a algunas de las situaciones más frecuentes.

### 4.4.1 Análisis con EnCase

EnCase Forensics es un producto comercial desarrollado por Guidance Software (<http://www.guidancesoftware.com>). Dispone de una interfaz fácil e intuitiva y no solamente se trata de una herramienta de gran potencia y versatilidad, sino que además, como ya se ha dicho en apartados anteriores, se trata de una aplicación que goza de un amplio respaldo oficial por parte de administraciones de justicia, cuerpos de policía y departamentos de seguridad de organizaciones y empresas de todo el mundo. Para sacar partido a EnCase es necesario hacer cursillos especializados y obtener una certificación patrocinada por la empresa desarrolladora (EnCase Certified Examiner o EnCe). No obstante el programa puede ser utilizado sin necesidad de disponer de conocimientos avanzados para algunas operaciones elementales de la investigación forense, como por ejemplo realización de imágenes a bajo nivel, recuperación de archivos borrados y rastreo de elementos de evidencia mediante búsquedas de caracteres y otras técnicas.

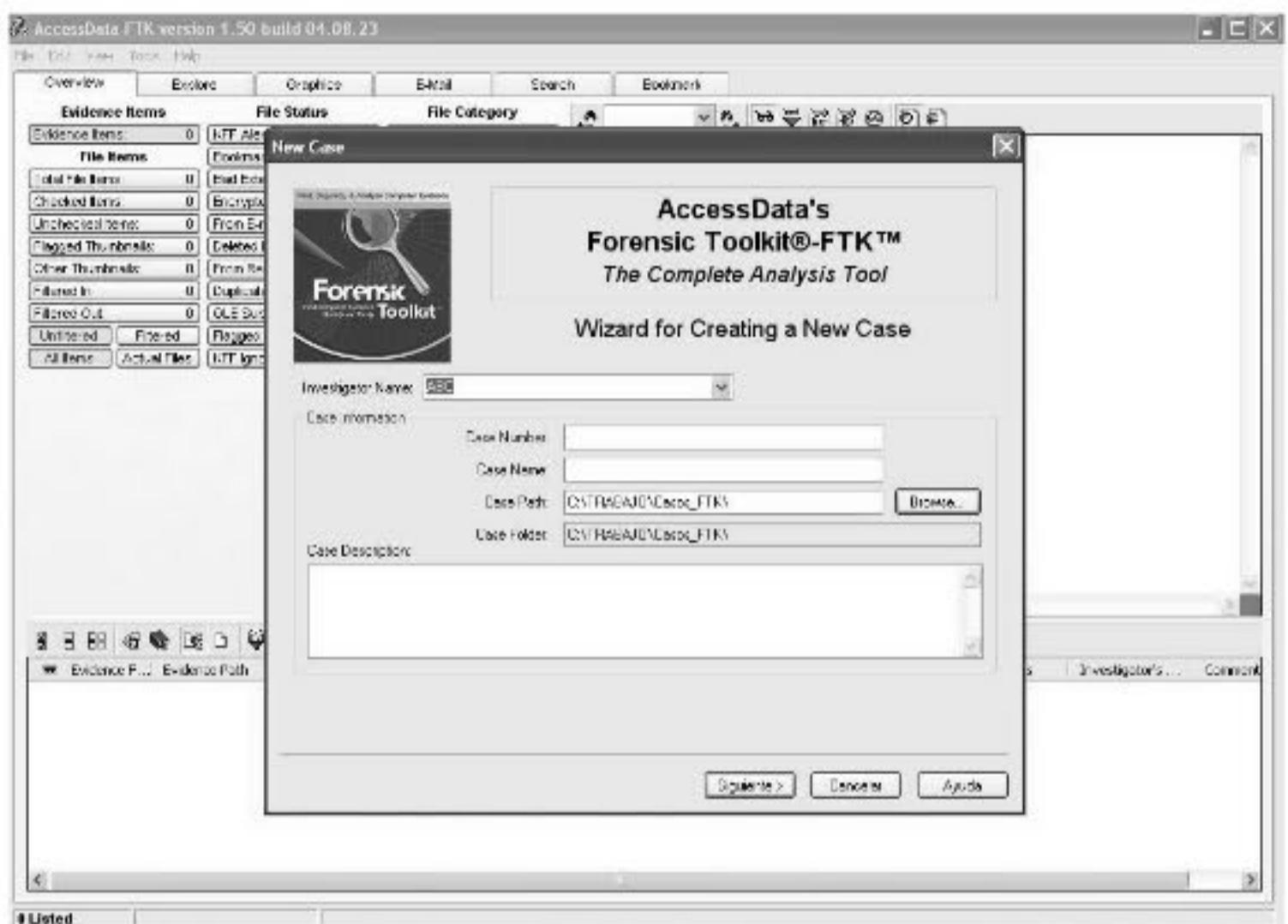


Figura 4.10. EnCase: interfaz de administración de casos

Para analizar una imagen adquirida —mediante dd o en el formato propietario de EnCase— lo primero que hay que hacer es añadirla al caso (en el menú Archivo). Una vez que el programa la detecta puede comenzar el análisis. EnCase permite al usuario navegar por directorios y examinar los espacios asignados por el sistema de archivos, incluyendo el *file slack*. Una funcionalidad importante consiste en la búsqueda de caracteres, lo cual resulta útil para localizar direcciones de correo electrónico, IP, URL, palabras clave y otros elementos de evidencia. La búsqueda incluye los sectores del volumen no asignados. Los resultados se muestran en forma de un índice ordenado por el que el investigador podrá navegar en busca de la información que le interesa. EnCase también permite recuperar archivos gráficos desde la caché borrada de Internet Explorer.

Una descripción detallada de EnCase y sus numerosas funcionalidades es algo que está más allá del alcance de esta obra. Si desea saber todo lo que se puede hacer con este software y por qué autoridades públicas y departamentos de seguridad de grandes empresas de todo el mundo lo tienen en tan alta estima el lector puede recurrir a una abundante documentación disponible en Internet o a la guía oficial de Sibex indicada en la bibliografía.

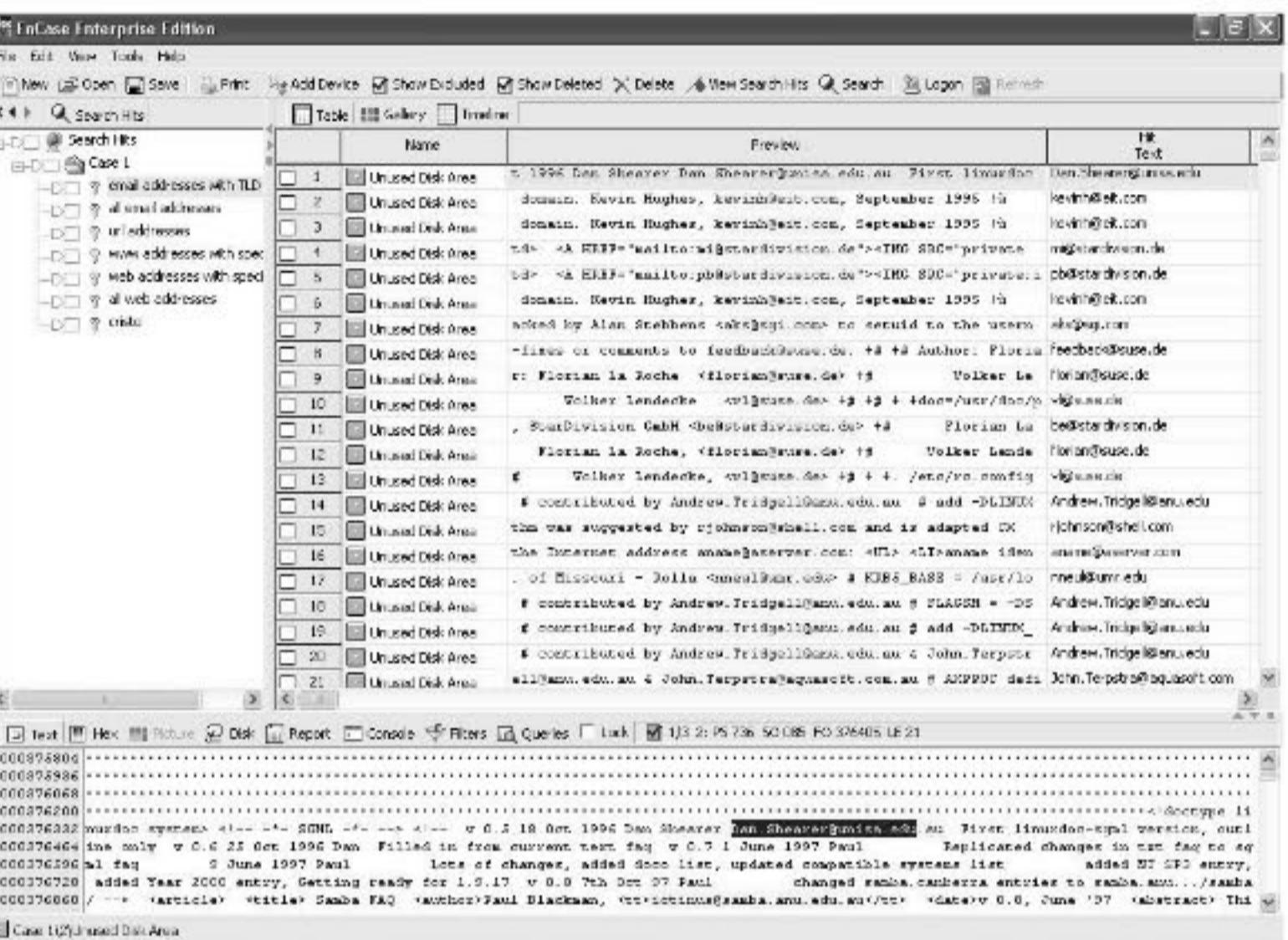


Figura 4.11. Análisis de elementos de evidencia con EnCase

#### 4.4.2 AccessData FTK

FTK también dispone de un interfaz de administración de casos (figura 4.12), pero el planteamiento es en este caso diferente. Mientras EnCase permite al investigador el acceso a todos los elementos del soporte de datos —sistema de archivos, espacio sin asignar, correo electrónico, historiales de Internet, etc.—, FTK, antes de que el investigador inicie su trabajo de análisis, lleva a cabo una exploración sistemática de la imagen para indexar todos los elementos que contiene. Esto no solo simplifica las posteriores tareas de investigación sino que también facilita la búsqueda mediante cadenas de caracteres y las hace más flexibles. Los nuevos soportes de datos pueden añadirse dinámicamente al caso según convenga, por ejemplo a medida que vaya ordenándose una incautación policial de los mismos. El acceso es en modo de solo lectura. También se extraen *hashes* MD5 o SHA de todos los archivos, con la doble finalidad de validar evidencias y permitir operaciones de filtrado KFF que reduzcan la carga de trabajo.

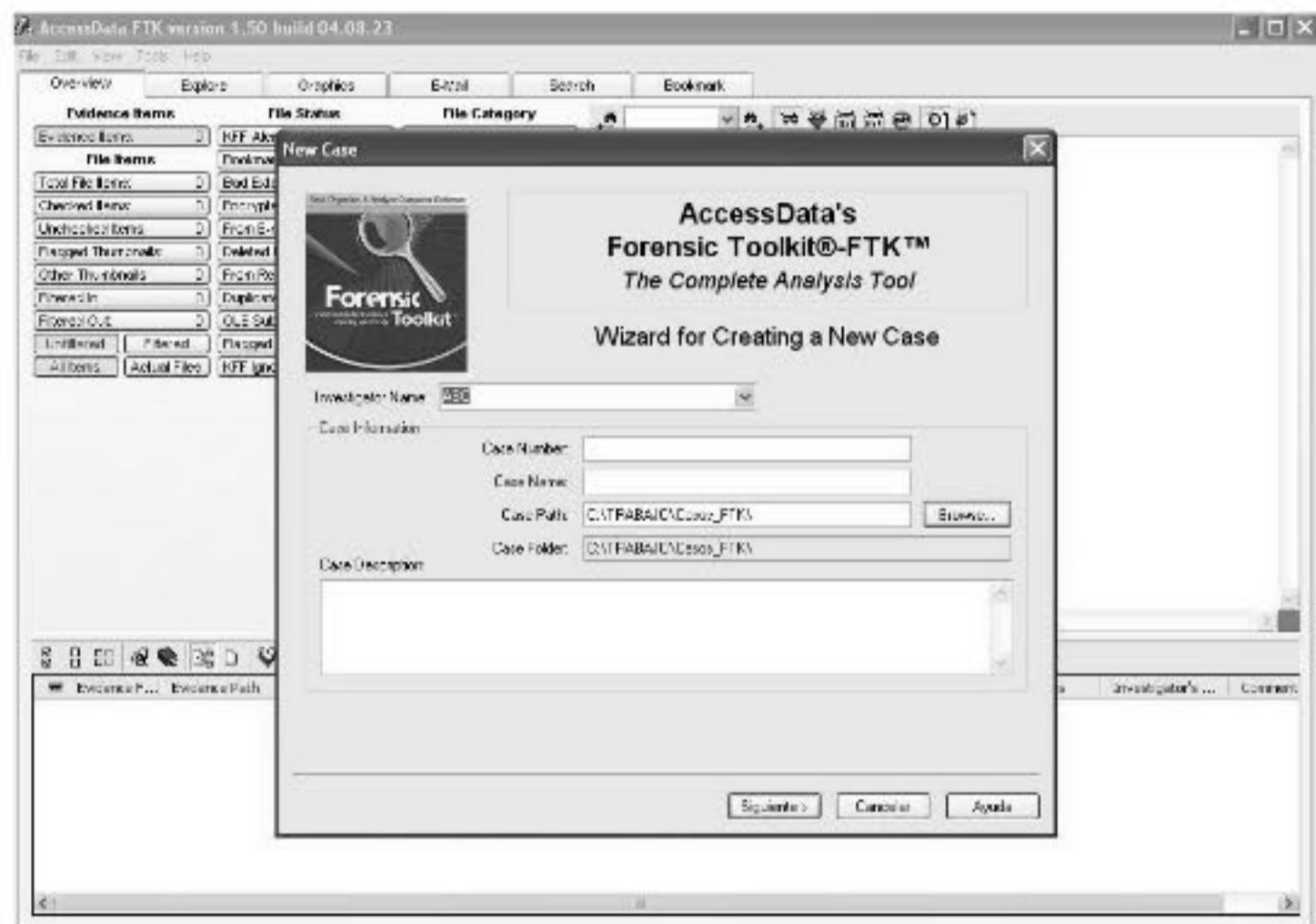


Figura 4.12. Administración de casos con FTK

El indexado es una operación muy intensiva en recursos del sistema. Al contrario que con EnCase, el investigador se verá obligado a esperar un buen rato mientras el ordenador completa las tareas de procesamiento inicial del caso – principalmente la indexación de contenidos del soporte de datos–, que para discos duros de gran volumen puede llegar a ser de varias horas. El resultado es un inventario completo de todos los elementos que contiene el soporte (figura XX). Durante el proceso AccessData FTK analiza los archivos y los va reuniendo por grupos. De modo simultáneo extrae sus *hashes* y los compara con la biblioteca KFF (*Known File Filter*) para averiguar si se trata de un archivo conocido perteneciente al sistema operativo o una aplicación de software estándar. De este modo la búsqueda puede concentrarse en los archivos que más interesan al investigador: documentos del usuario, troyanos, *logs* del sistema, mensajes de correo electrónico, etc.

Dentro del inventario, que aparece resumido en forma de tabla, el investigador puede saltar a la categoría que le interese. Los resultados se muestran en la parte inferior de la ventana. La recuperación de archivos borrados es de lo más simple y se lleva a cabo con el botón derecho del ratón, simplemente seleccionando un soporte de destino para salvarlos. La ventaja principal del indexado FTK consiste en la aceleración de las búsquedas de caracteres. Los

resultados se muestran de forma casi instantánea, lo cual facilita la tarea de investigación haciéndola más ágil y flexible. Al contrario que con EnCase y otras aplicaciones, no se necesita elaborar una lista exhaustiva de palabras y expresiones relacionadas con el caso, sino que el investigador puede ir probando sobre la marcha con lo que se le vaya ocurriendo, incluso con variantes de los términos, errores ortográficos y otras expresiones de tipo similar o relacionadas, etc.

Figura 4.13. Elementos de evidencia indexados por FTK

El análisis del espacio no asignado se realiza desde el menú **Slack → Free Space**. Un cuadro resumen muestra el número de referencias en cada categoría. Seleccionándolo resulta posible ver el contenido de los sectores no asignados. Para ello basta pulsar con el ratón en los registros que se muestran en la mitad inferior de la ventana. Cada uno de estos fragmentos puede ser extraído como archivo de texto o examinado en busca de cadenas de caracteres. FTK permite búsquedas con lógica difusa y expresiones regulares.

FTK también admite operaciones de tallado de archivos, lo cual resulta de gran utilidad para el análisis del espacio no asignado por el sistema de archivos y la búsqueda de elementos de evidencia en soportes formateados. La operación de *data carving* se lleva a cabo de manera sencilla e intuitiva a través de un interfaz en el menú de herramientas de la aplicación, que antes de ejecutar su tarea permite al usuario seleccionar los tipos de archivo que está buscando (gráficos, documentos

MS-Word, etc.). Los resultados se muestran en un listado debajo de un visor que permite examinar el contenido de cada archivo sin tener que utilizar el programa correspondiente (figura 4.14).

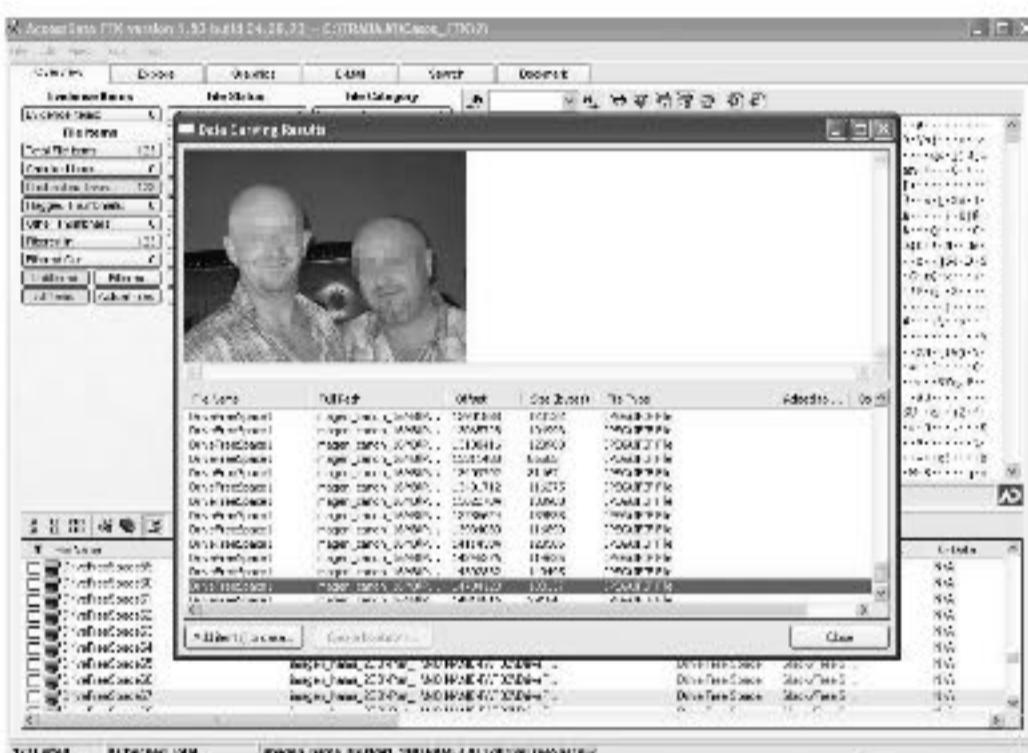


Figura 4.14. Data Carving con FTK

#### 4.4.3 Captain Nemo

Existen soluciones de bajo coste que no poseen la funcionalidad ni el automatismo de EnCase y FTK, pero permiten al investigador resolver uno de los objetivos fundamentales de su tarea: el acceso a imágenes en modo de solo lectura. Captain Nemo de la empresa RunTime (<http://www.runtime.org>) es una herramienta que posibilita el acceso a imágenes de medios montándolas de manera segura y sin enlazarlas con el sistema operativo.

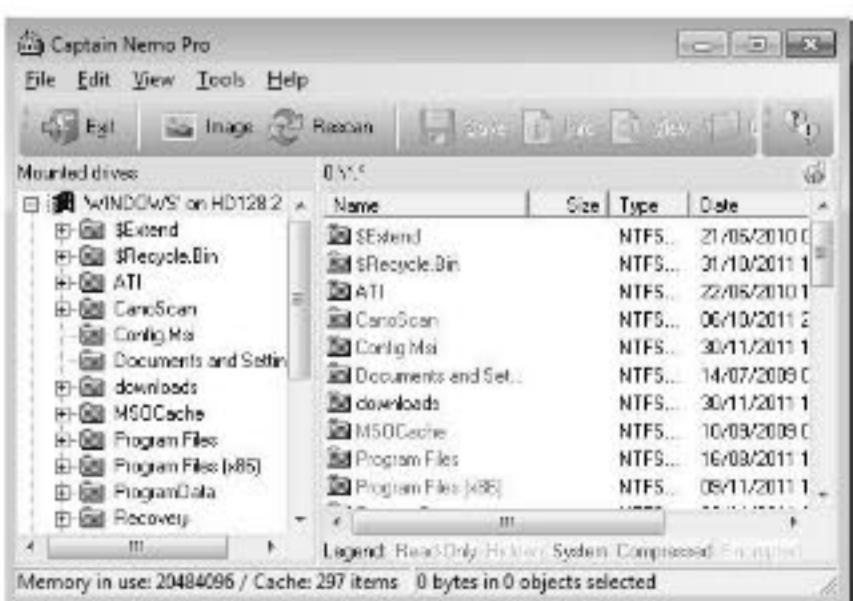


Figura 4.15. RunTime Captain Nemo

Además de las particiones típicas del entorno Windows NTFS y FAT también sirven para visualizar sistemas de archivos ext2 y Novell.

#### 4.4.4 Mount Image Pro

Mount Image Pro de GetData Pro es una herramienta diseñada para el montaje de imágenes de diversos formatos: dd, EnCase, e ISO, asignando a cada una de ellas una letra de unidad. Mount Image Pro monta las imágenes por defecto en modo de solo lectura y permite realizar la comprobación de los *hashes* MD5.

Las letras de unidad hacen posible acceder a los contenidos de la imagen forense a través de aplicaciones Windows. Se puede abrir archivos con las mismas utilidades ofimáticas con las que fueron creados o analizarlos en busca de código malicioso con el antivirus del ordenador. Y de igual modo resulta posible recuperar archivos borrados y exportar los resultados de la investigación.



Figura 4.16. GetDataBack Mount Image Pro

#### 4.4.5 FileDisk

FileDisk permite el montaje dinámico de sistemas de archivos en Windows de manera que se pueda acceder a las imágenes como si fueran unidades físicas. Esta herramienta, de la cual existen algunas variantes disponibles en la web del desarrollador (<http://www.acc.umu.se/~bosse/>), monta en modo de solo lectura una imagen obtenida con dd y permite analizar sus contenidos con cualquier herramienta forense Windows.

## 4.5 INVESTIGACIÓN DEL HISTORIAL DE INTERNET

Ya tenemos nuestra imagen binaria y los medios para acceder a ella. ¿Ahora qué? ¿Cómo sacar algo productivo de todo esto? No existe ninguna guía de instrucciones para llevar a cabo una investigación forense en el mundo real. En última instancia lo único que sirve de algo es la lógica y el sentido común. Es hora de que el lector, por supuesto únicamente a efectos operativos, deje a un lado las explicaciones teóricas y los diagramas de flujo. Como un buen criminalista deberá hacer un esfuerzo para ponerse en lugar del infractor y ver el mundo desde su perspectiva.

¿Qué hace la gente con sus ordenadores? ¿Dónde almacena sus archivos? ¿Cómo oculta la información comprometedora? Las respuestas dependen del puesto de trabajo, del software instalado y la pericia del sospechoso en el manejo de tecnologías de la información. En aplicaciones específicas como programas para control de sistemas industriales, diseño asistido por ordenador o gestión de almacenes las posibilidades de respuesta a las preguntas anteriores pueden ser de lo más variado. Sin embargo, cuando se trata de un PC convencional con Windows –situación que se da en el 90% de los casos– y las aplicaciones de productividad habituales, existen cauces más o menos definidos en los cuales el investigador podrá llevar a cabo su tarea utilizando estrategias y métodos estándar.

Los usuarios normales emplean sus ordenadores para escribir documentos y navegar por Internet. Esto último incluye la realización de gestiones bancarias, compras *on line* y correo electrónico a través de aplicaciones web como Gmail o Yahoo. También utilizan el ordenador para almacenar las fotos de sus cámaras digitales, descargar películas y audio en MP3, compartir archivos mediante utilidades como eMule o BitTorrent y –por supuesto– jugar. Los contenidos relacionados con todas estas actividades se guardan en ubicaciones que no tienen por qué ser necesariamente fijas, pero en la mayor parte de los casos vienen establecidas por defecto sin que el usuario se moleste en cambiarlas. En un ordenador sospechoso los datos que busca el investigador pueden estar colocados sin más en el sitio que les ha asignado por defecto el desarrollador del sistema operativo o de las aplicaciones de software. Pero nada impide que puedan encontrarse en otra ubicación del disco duro, ocultos, protegidos mediante encriptación o borrados para eliminar pruebas.

Quedó atrás el tiempo en que el ordenador se utilizaba para tareas estáticas como redactar informes, llevar la contabilidad de la empresa o manejar un entorno de programación. Ahora las telecomunicaciones y la Web forman parte esencial de la actividad del usuario en todos los ámbitos de la vida, tanto profesionales como particulares. Con el acceso universal a Internet y conexiones de banda ancha en la

mayor parte de las empresas y los hogares, el lugar más aconsejable para iniciar la investigación es el historial de Internet.

### 4.5.1 Microsoft Internet Explorer

Internet Explorer viene integrado en los sistemas operativos de Microsoft desde la primera versión de Windows XP. El procedimiento utilizado por Microsoft para guardar los archivos descargados durante la navegación –en carpetas que el sistema considera esenciales para su funcionamiento y mantiene ocultas al usuario– tiene un gran interés para el informático forense, tanto por la información que almacena como por el hecho de que resulta extremadamente difícil deshacerse de estos archivos con los métodos de borrado convencionales.

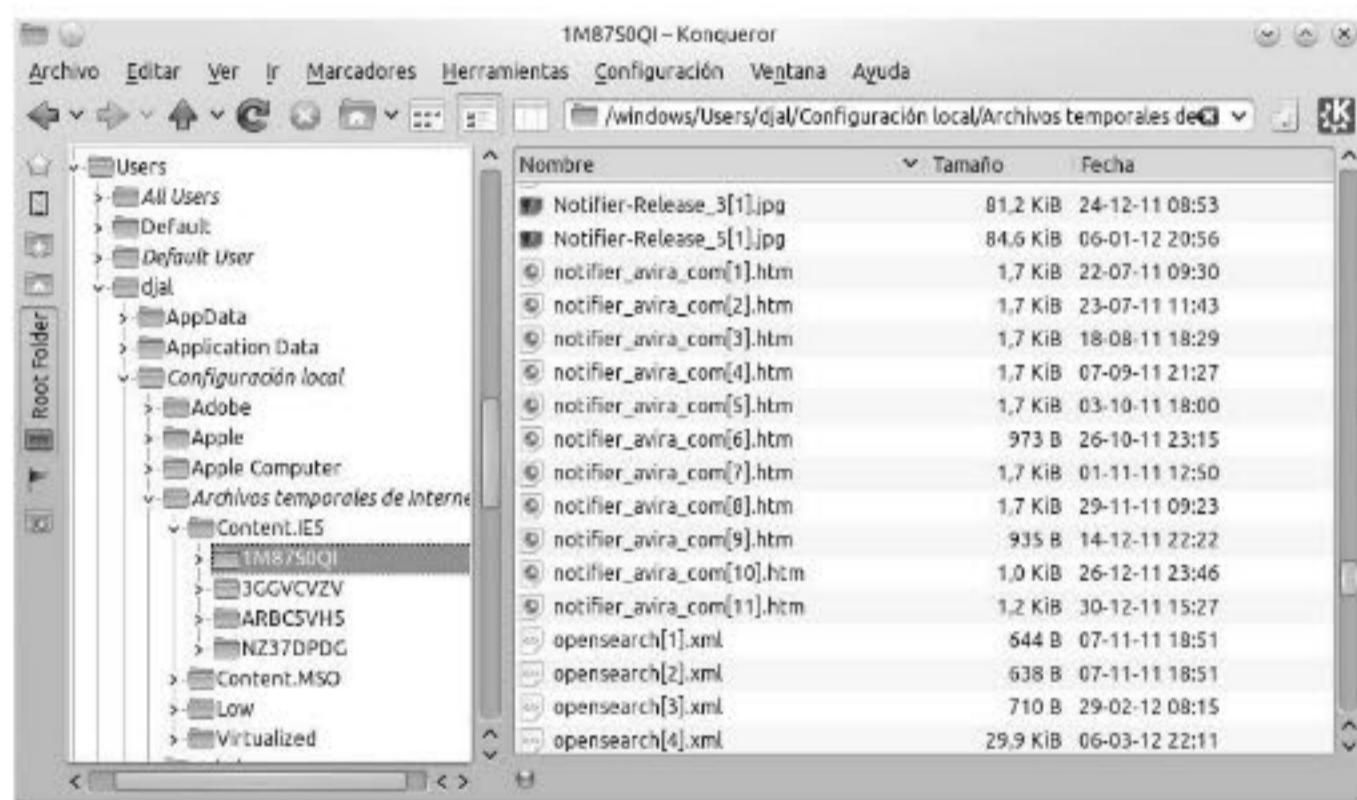


Figura 4.17. Ubicación del historial de Internet

Un caché es un sistema que los navegadores (Internet Explorer, Firefox, Opera o Safari) utilizan para guardar localmente el contenido de páginas web, archivos, imágenes y otros elementos. Esto evita tener que cargarlos de nuevo en visitas consecutivas, agilizando la experiencia de navegación del usuario. En Windows XP Internet Explorer almacena su caché en un árbol de directorios que permanece oculto al explorador de archivos. Esto es así porque se trata de una zona que el sistema operativo considera crítica para su funcionamiento. A no ser que sepa bien lo que hace, un usuario normal no debe manipular jamás estas estructuras de datos. El investigador podrá llegar hasta ellas a través de la consola de texto. Para ello vaya hasta el botón de **Iniciar**, seleccione **Ejecutar programa** y en la barra de comandos introduzca “cmd.exe”. Después trásladese hasta esta carpeta con el comando cambiar directorio (cd):

C:\Documents and Settings\<Nombre de usuario>\Configuración local\Historia\History.IE5

Montando la imagen adquirida con Mount Image Pro no tendrá ningún problema para llegar al caché de Internet, ya que en este caso, al no estar el sistema en funcionamiento no actuarán los mecanismos que impiden el acceso del usuario a las carpetas. También podrá acceder a ellas sin dificultad desde Linux (figura 4.17). La ubicación de la carpeta suele variar según la versión de MS-Windows o Internet Explorer, así como del idioma y otras características. En Windows 7, por ejemplo, se encuentra en:

C:\Users\<Nombre de usuario>\Configuración local\Archivos temporales de Internet\History.IE5

Dentro de este directorio el investigador hallará diversas carpetas con gran cantidad de archivos resultantes de la actividad del usuario en Internet: imágenes, documentos, páginas web, código java script, etc. Lo que nos interesa es un archivo llamado INDEX.DAT, que contiene el historial de navegación con las páginas visitadas en las fechas y horas que se indican y otros elementos de interés. El archivo INDEX.DAT no es de texto sino de tipo binario. Para examinar su contenido necesitamos herramientas como Pasco (<http://www.foundstone.com/us/resources/proddesc/pasco.htm>) o strings de Sysinternals. Pasco permite obtener un archivo csv con el historial de Internet en forma de tabla que después se podrá examinar con Excel o con Open Office:

C:\> pasco index.dat > historial.csv

#### 4.5.2 X-Ways Trace

Con X-Ways Trace, una herramienta desarrollada por la empresa alemana X-Ways Software AG, se puede analizar el archivo Index.dat del caché interno de Internet Explorer. Presenta los resultados –URL completa, fecha del último acceso, identificación del usuario, tamaño de archivo, etc.– en una tabla con campos cuyos datos podrán ser importados posteriormente a una hoja de cálculo Excel. X-Ways Trace decodifica archivos history.dat con el historial de Mozilla/Firefox y dcache4.url típicos del navegador Opera, y también el archivo Info2 de la papelera de Windows (en versiones inferiores a Vista).

Se trata de un software anticuado pero todavía utilizable. Su última versión disponible data del año 2008 y no existe garantía de que funcione con los navegadores recientes. Existe una versión de prueba gratuita en la página web de X-Ways (<http://www.x-ways.net>). Se puede utilizar libremente, con la única restricción de que no es plenamente operativa y los resultados que se muestran están limitados a las 500 primeras entradas del historial de navegación.

Figura 4.18. X-Ways Trace

#### 4.5.3 iehist

Si lo que se desea es conocer de modo rápido las páginas de Internet que han sido visitadas desde un ordenador sospechoso se puede recurrir a la herramienta iehist (<http://www.cquare.net/wp/iehist/>), gratuita y de manejo simple:

C:\> iehist.exe index.dat > historial.txt

#### 4.5.4 Historial de navegación en Mozilla/Firefox

En la actualidad el navegador Mozilla/Firefox es el más utilizado después de Internet Explorer, y lleva camino de superarlo de aquí a pocos años. A la fecha de escribir estas líneas al menos un 25% de los usuarios lo usan para su actividad habitual en Internet. Existen versiones para Windows, Linux y Mac OSX. Pese a la copiosa documentación y la transparencia del código libre, existen aún pocas herramientas para el análisis automatizado del historial de navegación de Internet en Firefox.

Firefox almacena toda su información, incluyendo el historial de Internet, en bases de datos SQLite. Por tanto es necesario que el investigador se familiarice con este formato y las herramientas utilizadas para gestionarlo, afortunadamente abundantes y casi todas ellas de código libre. En Windows los archivos que

interesa localizar se encuentran en ubicaciones características dependiendo de la versión del sistema operativo.

En Windows XP:

C:\Documents and Settings\%username\Configuración local\Aplicaciones\Mozilla\Firefox\Profiles

En Windows Vista/7:

C:\Users\%username\AppData\Roaming\Mozilla\Firefox\Profiles

En este directorio hay un archivo denominado profiles.ini que en el arranque de Firefox determina el directorio del cual el navegador debe leer los datos necesarios para configurar la sesión. También se encuentran allí todos los archivos en los que se guarda la información relativa a la actividad correspondiente al perfil de un usuario determinado. Cuatro de ellos son de particular interés para el investigador forense:

- **Formhistory.sqlite:** incluye datos referentes a entradas de formularios, recuadros de búsqueda, nombres de usuario, etc.
- **Downloads.sqlite:** archivos descargados por el usuario.
- **Cookies.sqlite:** información relativa a *cookies*.
- **Places.sqlite:** aquí es donde se encuentra la masa principal de datos relacionada con la actividad en Internet y las páginas visitadas.

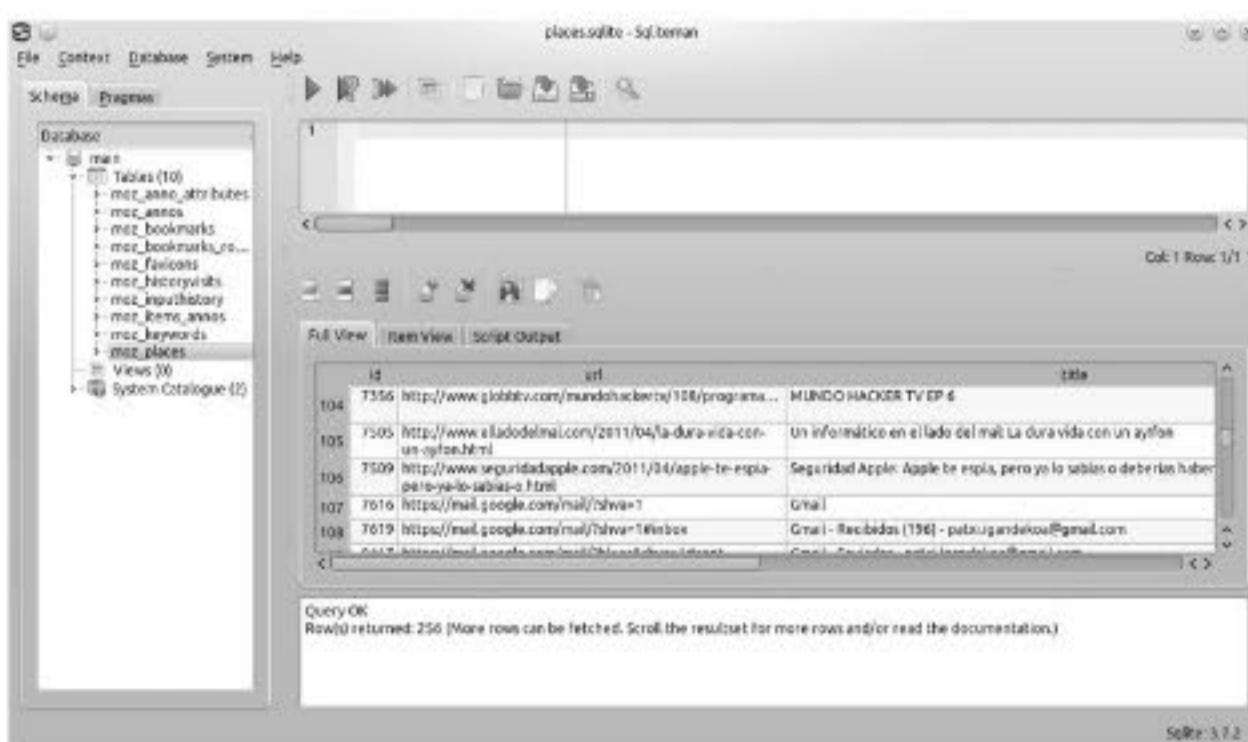


Figura 4.19. Investigación del historial de Internet de Firefox con SQLiteman

SQLiteman es una aplicación de código libre con interfaz gráfica que permite al usuario interactuar con bases de datos SQLite. Para trabajar con él solo hace falta recuperar la carpeta con los perfiles de Firefox, consultar el archivo profiles.ini y abrir los archivos con extensión .sqlite que interesen. Las bases de datos de Firefox poseen estructura simple, con una sola tabla de interés en la mayor parte de los casos.

No olvide que los clientes SQLite no fueron desarrollados para la investigación forense, sino para la gestión y el mantenimiento de datos. Con un cliente SQLite no solo se puede leer información, sino también modificarla. Por este motivo y para evitar cualquier alteración accidental de datos el análisis de historiales de navegación de Mozilla Firefox deberá llevarse a cabo siempre en duplicados y jamás sobre el soporte original.

Además del historial de navegación, otro lugar interesante para investigar es el caché de Firefox. Se encuentra en la carpeta de Perfiles y contiene gran cantidad de archivos, casi todos ellos binarios, relacionados con la exploración de páginas web y sitios de Internet. Hay aplicaciones forenses que permiten interpretar estos archivos y extraer de ellos información que puede resultar útil para nuestra investigación.

#### 4.5.5 Chrome

Otro navegador de Internet que adquiere cada vez más relevancia, quitando protagonismo a Internet Explorer, aunque no a la misma velocidad que Firefox, es Google Chrome. Al igual que Firefox, Chrome utiliza bases de datos SQLite para organizar los datos generados por la actividad del usuario. He aquí los emplazamientos de los archivos:

En Windows XP:

C:\Documents and Settings\%username\Configuración local\Aplicaciones\Google\Chrome\default

En Windows Vista/7:

C:\Users\%username\AppData\Local\Google\Chrome\default

#### 4.6 LA PAPELERA DE RECICLAJE

Todos los usuarios están familiarizados con la papelera de Windows: cuando se borra un archivo este no es eliminado del disco duro, sino trasladado a un almacén en el que se guarda de manera provisional por si el usuario necesita recuperarlo. Finalmente, si no interesa conservar el archivo, se puede eliminar definitivamente vaciando la papelera de reciclaje. El carácter definitivo de

esta eliminación ha de entenderse con las mismas reservas aplicables al borrado de archivos por métodos convencionales y a la posibilidad de que los mismos sean recuperados aplicando las técnicas forenses descritas en el capítulo 3.

La papelera funciona del modo siguiente: el usuario marca un archivo pulsando sobre él con el botón derecho del ratón. Un menú contextual le pregunta, entre otras opciones, qué desea hacer con el archivo, si eliminarlo definitivamente o trasladarlo a la papelera. Si el usuario elige esto último el sistema traslada el archivo a la papelera quitándole su nombre original y asignándole otro con este formato:

D <Unidad de Windows> <Número consecutivo>. Extensión del archivo

```
C:\> dir/a info2
El volumen de la unidad C no tiene etiqueta.
El n mero de serie del volumen es: 0C7D-0A68
Directorio de C:\RECYCLER\S-1-5-21-73586283-1682526488-854245398-1003>
14/04/2012 12:15 419.220 INFO2
 1 archivos 419.220 bytes
 0 dirs 5.541.294.080 bytes libres
C:\>
```

Figura 4.20. Carpeta de la Papelera de Windows con el archivo INFO2

Los cambios son registrados por el archivo INFO2, que en sistemas del tipo Windows NT/XP/2003 se encuentra en C:\Recycler\<Identificador de usuario>\INFO2 (figura 4.19). Este archivo es generado automáticamente por el sistema operativo tan pronto como el primer archivo se borra y va a parar a la papelera. Al vaciarla, el contenido de INFO2 se elimina y la numeración comienza otra vez consecutivamente desde el principio para los archivos borrados con posterioridad.

La papelera de Windows es una carpeta protegida del sistema. Ni siquiera puede verse a través del explorador de archivos. La única manera de llegar al archivo INFO2 es mediante la consola de texto. INFO2 es invisible y no se muestra en un listado de directorio normal. Para verlo deberá utilizar el comando dir/a. Una vez en su poder puede examinarlo directamente o copiarlo a otra ubicación. Si decide hacer esto último, quizás se encuentre con que al tratarse de un archivo protegido el comando copy no le sirve de nada. Pruebe con un pequeño truco de redirecciónamiento:

C:\> RECYCLER\[Id. De usuario]> type INFO2 > c:\trabajo\info2

Si la imagen está montada con Mount Image Pro o en Linux no tendrá restricciones de ningún tipo para acceder a ella, ni tampoco para copiar el archivo INFO2 al lugar de su estación de trabajo que más le convenga.

#### 4.6.1 Análisis de la papelera con Rifiuti

Rifiuti (palabra italiana que significa “basura”) es una herramienta de MacAfee (<http://www.mcafee.com/es/downloads/free-tools/rifiuti.aspx>) que permite interpretar de modo comprensible el contenido del archivo INFO2. La sintaxis de empleo es simple:

rifiuti INFO2 > papelera.txt

Mediante redireccionamiento podemos crear un archivo de texto o con extensión CSV (acrónimo para estructuras de datos simples en forma tabulada y separados por comas) que podremos examinar con el Bloc de notas, Excel o cualquier otro programa de usuario, y en el que se mostrarán todas las informaciones pertinentes a cada uno de los archivos borrados: nombre, fecha de eliminación, unidad, tamaño, etc.

#### 4.6.2 Funcionamiento de la papelera en Windows Vista/7

En los sistemas Windows recientes –Vista y 7– la Papelera de reciclaje ha sido sometida a una labor de rediseño que hace imposible la utilización de las herramientas mencionadas con anterioridad. Para empezar ya no hay ningún archivo INFO2, sino que por cada archivo que el usuario elimina se producen otros dos:

\$I <Secuencia aleatoria de caracteres> .[extensión del archivo original]  
\$R<Secuencia aleatoria de caracteres> .[extensión del archivo original]

Supongamos que el usuario del sistema sospechoso tiene un archivo Excel denominado Caja\_B.xlsx y lo borra sin más. Para encontrarlo vamos a la carpeta en la que se aloja la papelera –en Windows Vista y 7 un directorio de sistema oculto llamado \$Recycle.bin <Identificador de usuario>– y examinamos su contenido. Los dos archivos correspondientes a la evidencia podrían ser algo parecido a esto:

\$IADT5FR.xlsx  
\$RADT5FR.xlsx

El archivo renombrado que comienza por \$I contiene la información necesaria para recuperar el archivo original, incluyendo el nombre y la ruta completa en el árbol de carpetas. Obsérvese que todos los archivos de este tipo

tienen la misma longitud: 544 bytes, y pueden ser examinados sin problemas a través del Bloc de notas. En el segundo archivo se encuentran los datos en bruto: texto, imágenes, sonido, hojas de cálculo, etc.

## 4.7 COOKIES

Las *cookies* son los artefactos más polémicos de Internet. La literatura técnica y los medios las relacionan ampliamente y de modo a veces sensacionalista con fallos de seguridad —secuestro de sesiones, robos de identidad y seguimiento de usuarios, *spam* y otras operaciones ilícitas—. Sin embargo resultan imprescindibles para gran variedad de tareas, sin las cuales no sería posible la web interactiva que hoy conocemos.

La comunicación entre un navegador y un servidor de páginas web se lleva a cabo mediante HTTP, un protocolo sin estado que no es capaz de recordar los parámetros de conexión. El navegador —Internet Explorer, Mozilla Firefox, Opera, Chrome— envía una petición; el servidor web la recibe, manda la página correspondiente, termina la conexión y queda a la espera. Siempre que el navegador necesita una página se repite el proceso. Esto parecía razonable en los primeros tiempos, cuando la web estaba compuesta principalmente por páginas estáticas vinculadas a través de hiperenlaces, pero para la forma en que se utilizan hoy en día los recursos de Internet resulta enormemente farragoso. Imagine que un usuario accede a su correo electrónico a través de Gmail. Si cada vez que quisiera pasar del buzón de entrada a la carpeta de los elementos borrados o al formulario de correo nuevo tuviese que introducir nuevamente un nombre de usuario y una contraseña su experiencia de navegación podría llegar a convertirse en un auténtico fastidio.

El servidor soluciona este problema colocando un pequeño archivo de texto —o *cookie*— en una ubicación determinada del ordenador del usuario (en MS-Windows 2000/XP/2003: C:\Documents and Settings\<Nombre de usuario>\Cookies). Cada vez que un navegador se conecta a una página principal o solicita un recurso del sitio web, lo primero que hace es examinar la carpeta de *cookies* en busca de alguna que haya quedado allí como resultado de una visita anterior. De ser el caso, a partir de los datos contenidos en la *cookie* el servidor recupera la información de estado y las preferencias del usuario. De este modo estará en condiciones de guiarlo hasta la parte del sitio web por la que acostumbra a moverse, recordar sus preferencias de idioma o visualización e incluso permitir el ingreso del usuario sin haberse identificado ni introducido una contraseña, en caso de que lo hubiera ya hecho con anterioridad y la sesión continuara estando abierta.

Básicamente lo que hace el servidor es marcar al cliente para saber que permanece conectado, si lo hizo con anterioridad, o si viene de otro sitio de Internet con el que el servidor mantenga algún intercambio de información. Las *cookies* permiten al usuario acceder a su aplicación de correo web sin necesidad de identificarse aun después de haber permanecido desconectado de Internet durante varios días. También hacen posible el funcionamiento de carritos de la compra en todo tipo de tiendas *on line*, y con ello uno de los principales elementos del correo electrónico. El enjuiciamiento de las *cookies* como amenaza para la privacidad del usuario no constituye objeto de la presente obra. Lo único que importa aquí es su valor como elemento de evidencia forense. La presencia de una *cookie* puede ser indicativa de las páginas visitadas y otras actividades en Internet.

Las *cookies* carecen de formato estándar. Cada sitio de Internet las diseña como más le conviene. Normalmente se almacenan en memoria RAM hasta que se cierra el navegador, momento en que son archivadas en el disco duro para quedarse allí durante un período que depende de la fecha de caducidad asignada por el servidor, o hasta que el usuario se deshaga de ellas borrándolas. Las *cookies* pueden visualizarse a través de cualquier editor de texto. Existe una aplicación para analizarlas en bloque: *galleta*. Se trata de una herramienta desarrollada por Foundstone capaz de generar un informe en formato csv que posteriormente puede ser examinado con cualquier software de hojas de cálculo como Excel.

```
galleta [opciones] <Nombre de archivo>
-t separador de campo (por defecto: tabulador)
```

Ejemplo de utilización:

```
galleta usuario@cookie_misteriosa.txt > resultados.txt
```

## 4.8 CORREO ELECTRÓNICO

El correo electrónico es uno de los objetivos principales de la investigación forense. Conviene aclarar que no estamos hablando del que se gestiona a través de aplicaciones web como Yahoo o Google Mail, sino del correo electrónico almacenado localmente tras haberlo descargado con un cliente Outlook, Eudora o de cualquier otro tipo mediante los protocolos POP3 o IMAP. Lo ideal sería examinarlo con la misma aplicación que el sospechoso utiliza para escribir, recibir y almacenar sus mensajes. Sin embargo en la práctica también es necesario recurrir a herramientas de terceros. Tanto en un caso como en el otro conviene tener nociones relativas al funcionamiento y la estructura interna de los formatos de archivo habituales como MS-Outlook, Eudora o la cada vez más popular solución de código libre Thunderbird.

#### 4.8.1 Formatos PST y DBX Folders

Outlook y Outlook Express son los clientes de correo electrónico más utilizados por los usuarios de Windows. Ambos utilizan un formato propietario de archivos binarios muy incómodo de visualizar a través de editores de texto. Existen sin embargo herramientas capaces de interpretar la estructura interna y los contenidos de los archivos PST de Outlook, como por ejemplo PST-Viewer (<http://www.pstviewer.com/es/>) de Encryptomatic. PST-Viewer (figura 4.21), además de examinar archivos PST de Outlook, permite abrir mensajes de correo electrónico MSG y objetos incrustados OLE sin necesidad de tener instalado MAPI ni el cliente de correo electrónico Outlook.



Figura 4.21. Encryptomatic PST-Viewer

DBX Folders es un formato algo más complejo que PST utilizado por Microsoft para guardar el correo electrónico Outlook Express. Hay dos tipos de archivos DBX. El primero se denomina Folders DBX, y consiste en un catálogo de los restantes archivos DBX pertenecientes a un usuario particular del sistema. El segundo tipo es E-Mail DBX, donde se almacenan los mensajes de correo electrónico y los archivos adjuntos. Cada archivo del tipo E-Mail DBX se encuentra catalogado en el archivo Folders DBX de modo que Outlook Express pueda reconstruir el árbol de carpetas de correo electrónico correspondiente al usuario: buzón de entrada, elementos enviados, borradores y mensajes borrados.

Cada archivo Folders DBX es específico de un usuario del sistema y por defecto se halla localizado en la carpeta de datos de aquél, que en Windows XP, por ejemplo, se denomina C:\Documents and Settings\Usuario\Configuración local\Datos de aplicaciones\<Código de usuario>\Microsoft\Outlook Express. Para analizar su estructura interna se puede utilizar Eindeutig, una herramienta de código libre desarrollada por Keith J. Jones y disponible en <http://www.jrdcorp.com/index.php/blog-mainmenu-46/44-tools/85-eindeutig>. Tecleando eindeutig en línea de comando obtendremos un mensaje que nos informa sobre el modo de uso y las diferentes opciones.

#### 4.8.2 Otros clientes de correo

Eudora, una de las soluciones de correo electrónico más ampliamente utilizadas después de Microsoft Outlook, y Thunderbird, que adquiere protagonismo gracias al tirón de Mozilla Firefox y el código libre, no plantean excesiva dificultad a la hora de examinar evidencias basadas en *e-mail*. Estos programas guardan el correo en archivos de texto de tipo .mbx o .mbox que pueden ser examinados con cualquier visor e incluso con procesadores de texto normales Open Office o MS-Word. Basta con abrir el archivo como si fuera un documento normal. Las búsquedas de cadenas de caracteres se pueden llevar a cabo desde Wordpad o el propio procesador de textos.

#### 4.8.3 Paraben's E-Mail Examiner

Si el investigador trabaja para la Policía, las Fuerzas Armadas o el departamento de seguridad de una gran empresa, puede convencer a sus superiores para que autoricen la compra de una herramienta que facilita en gran medida la labor de análisis de los formatos de correo electrónico existentes. Será una buena inversión. Con Paraben's E-Mail Examiner, de Paraben Corporation, no solamente podrá visualizar el correo, sino también examinar la estructura de los archivos en los que se guardan los mensajes, adjuntos y páginas de formato en HTML. Así mismo podrá analizar la evidencia aplicando criterios de clasificación y búsqueda de caracteres. Paraben's E-Mail Examiner ofrece una funcionalidad de recuperación de mensajes borrados –entendiendo por tales no los que el usuario traslada a la papelera del correo electrónico, sino también los que se creía que estaban eliminados definitivamente, y que en virtud de la dinámica de borrado de archivos explicada en apartados anteriores resultan estar aún vivos en algún lugar del sistema o del espacio no asignado del soporte de datos–. Existe una versión de prueba disponible en <http://www.paraben.com/email-examiner.html>.

## 4.9 BÚSQUEDA DE CARACTERES

Una investigación forense que se conforma con un inventario de documentos y el rescate de unos cuantos archivos borrados es una investigación a medias. Para llegar hasta el fondo del caso es preciso emplear algo más de esfuerzo y agotar el rango de posibilidades mediante la búsqueda de elementos de evidencia por cadenas de texto. El objetivo consiste en localizar texto revelador de las actividades delictivas llevadas a cabo por el sospechoso mediante palabras clave o expresiones regulares. Una expresión regular no es otra cosa que una cadena de texto que se ajusta a un formato determinado y en la que algunos caracteres o todos ellos han sido reemplazados por comodines. Por ejemplo, si estamos buscando números de teléfono en España, sabemos que las cadenas han de tener once caracteres decimales del 0 al 9 y uno de estos formatos: +34-XXX XXX XXX; 0034-XXX-XXX-XXX; +34 XXX.XX.XX XX. Si por otro lado el objetivo del investigador consiste en localizar números de tarjeta de crédito, deberá afinar su interfaz de búsqueda para que localice secuencias de veinte caracteres numéricos del 0 al 9 separados por rayas o por espacios de tabulador, con dos campos de caracteres vecinos, uno con números del 1 al 12 y de dos cifras separados por una raya oblicua para las fechas de caducidad y otro de tres cifras para los dígitos de control, y así sucesivamente.

Esta fase de la investigación parece intuitiva y trivial. En realidad la búsqueda de caracteres resulta bastante compleja, ya que su éxito implica una preparación adecuada y esfuerzos de lógica y racionalización que tengan en cuenta todas las circunstancias que pueden dificultar la tarea, como por ejemplo la codificación de caracteres. Si la búsqueda se lleva a cabo sobre caracteres Unicode o idiomas extraeuropeos, una configuración incorrecta de la herramienta de búsqueda hará fracasar la investigación. Para obtener buenos resultados y ahorrar tiempo, que con la proliferación de casos relacionados con soportes digitales y el incremento subsiguiente en la carga de trabajo del personal investigador lleva camino de convertirse en el recurso más valioso de los laboratorios forenses, se requiere una estrategia planificada y la confección de listas de palabras y expresiones regulares adecuadas.

### 4.9.1 SectorSpy, Disk Investigator y Evidor

Utilidades como SectorSpy, Disk Investigator y Evidor nos permiten, literalmente, rastillar los sectores de un disco duro en busca de cadenas de caracteres. La búsqueda se lleva a cabo a bajo nivel sin recurrir a las rutinas Windows que gestionan el sistema de archivos. Disk Investigator permite navegar por directorios y carpetas, identificar archivos borrados e incluso recuperarlos.

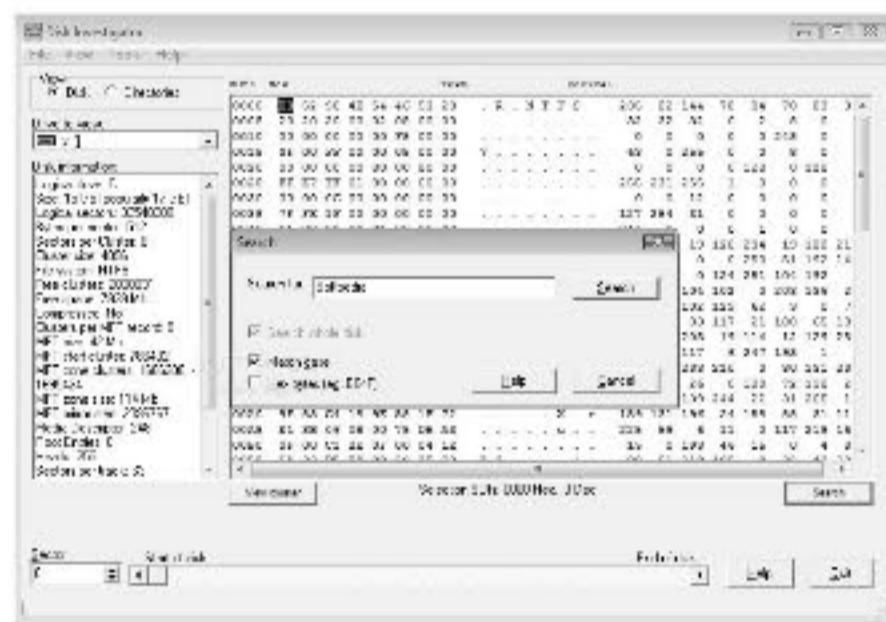


Figura 4.22. Disk Investigator

Disk Investigator (figura 4.22) es una herramienta de código libre, y su reducido tamaño (aproximadamente 500 KB) la convierten en una solución interesante para casos de emergencia y entornos de investigación portables.

### 4.9.2 X-Ways Forensics

Otra herramienta para la localización de cadenas de caracteres en un soporte de datos es Evidor de X-Ways Forensics (<http://www.x-ways.net/evidor>). Evidor, que emplea las funciones de búsqueda a bajo nivel del editor hexadecIMAL WinHex, proporciona una visión global y comprensible tanto de las unidades lógicas como del medio físico de datos. También cuenta con una rutina automática de elaboración de informes que anota la posición de las cadenas de caracteres localizadas.

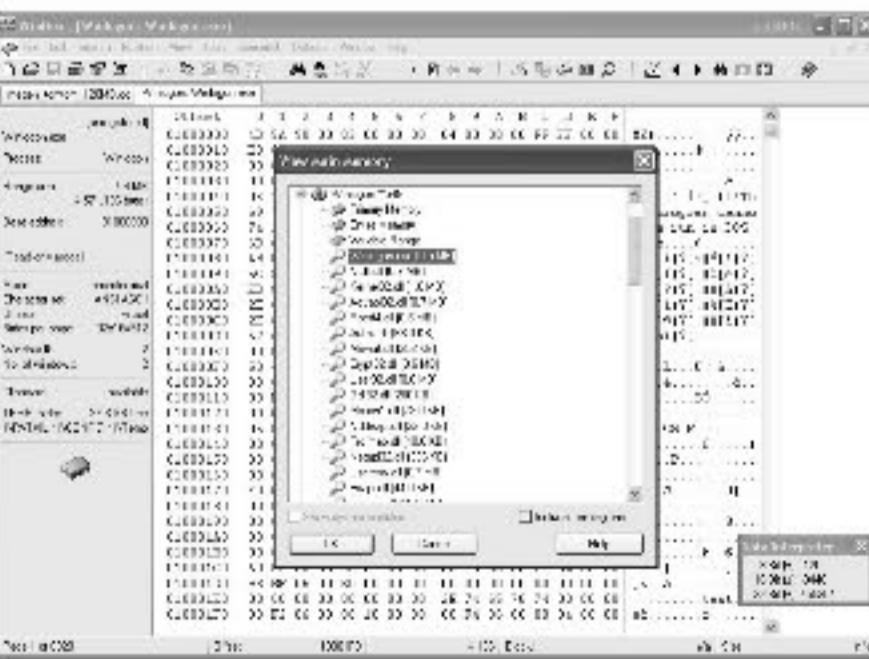


Figura 4.23. WinHex examinando un proceso del sistema en RAM

X-Ways Forensics, más que una *suite* de herramientas, es un conjunto de utilidades desarrollado por la empresa alemana X-Ways Software Technology AG, que trabaja como proveedor de soluciones de seguridad e Informática Forense para un número de casas importantes del sector de las Tecnologías de la Información (entre ellas Microsoft, Hewlett Packard, Novell, Toshiba, etc.). Uno de sus productos principales es Win-Hex (figura 4.23), sofisticado visor hexadecimal que además de las funcionalidades típicas de estas aplicaciones –examen de archivos y sectores de disco a bajo nivel– posee otras complementarias que en manos expertas la convierten en una herramienta de gran potencia. Con WinHex se pueden realizar búsquedas por cadenas tanto de texto como de código hexadecimal, realizar operaciones de *data carving* y rescate de archivos borrados e incluso analizar los procesos que se están ejecutando en RAM.

## 4.10 METADATOS

Los metadatos constituyen una fuente de información de gran valor para el investigador forense. De modo simplificado podemos describirlos como datos que hacen referencia a otros datos. Se trata de un nivel extra de información que se genera y se añade a un archivo informático de manera automática. Cuando el usuario crea y edita documentos de un tipo determinado (archivos MS-Office, documentos Adobe PDF, imágenes, material audiovisual, etc.) automáticamente está produciendo información sobre su actividad en el sistema. En su página web Microsoft advierte que las aplicaciones Office (Word, Excel, PowerPoint) crean metadatos de los tipos que se indican a continuación:

- Nombre e iniciales del usuario (o de la persona que crea el documento).
- Empresa o entidad.
- Nombre del ordenador dentro de la red.
- Carpeta que contiene el documento dentro del disco duro o del servidor.
- Impresora utilizada para obtener copias en papel.
- Otras propiedades del documento y resumen del contenido.
- Porciones no visibles de objetos OLE incrustados.
- Revisiones sucesivas del documento, incluyendo texto borrado.
- Autores anteriores del documento.
- Versión del procesador de texto (MS-Word 9.0, 10.0, etc.).
- Sistema operativo utilizado.

- Plantillas utilizadas para crear el archivo.
- Texto oculto.
- Comentarios.

Los metadatos resultan útiles para clasificar documentos de modo que puedan ser localizados fácilmente en repositorios de archivos. Estos documentos, además de su contenido explícito en forma de texto, tablas e imágenes tal y como se muestran en pantalla, llevan consigo una información incrustada que no se aprecia a simple vista, pero a la cual podemos acceder a veces por un procedimiento tan trivial como abrir los archivos en el Bloc de notas de Windows o con un editor hexadecimal. Con frecuencia estos datos ocultos consisten en información reservada que puede resultar perjudicial en manos de un destinatario equivocado o al ser vista por personas no autorizadas. El investigador forense los utilizará para la búsqueda de elementos de evidencia y sus tareas de análisis.

### 4.10.1 Cómo visualizar los metadatos de un documento

Puede haber metadatos en cualquier lugar del documento, dependiendo del programa utilizado para crearlo. Para llegar hasta ellos en principio no hace falta ninguna herramienta de software. Abriendo el archivo con un editor hexadecimal como WinHex o HxD (figura 4.24) se podrán ver sin dificultad. La mayor parte de las veces la información que busca aparece como texto claro o caracteres Unicode fácilmente distinguibles del ruido de fondo binario. Guíese por la columna derecha del editor, en la que se muestra el contenido del archivo en caracteres ASCII.

Este método resulta poco práctico si tenemos que manejar grandes cantidades de archivos o documentos extensos. En tales circunstancias es necesario recurrir a una aplicación que localice y clasifique los metadatos automáticamente.

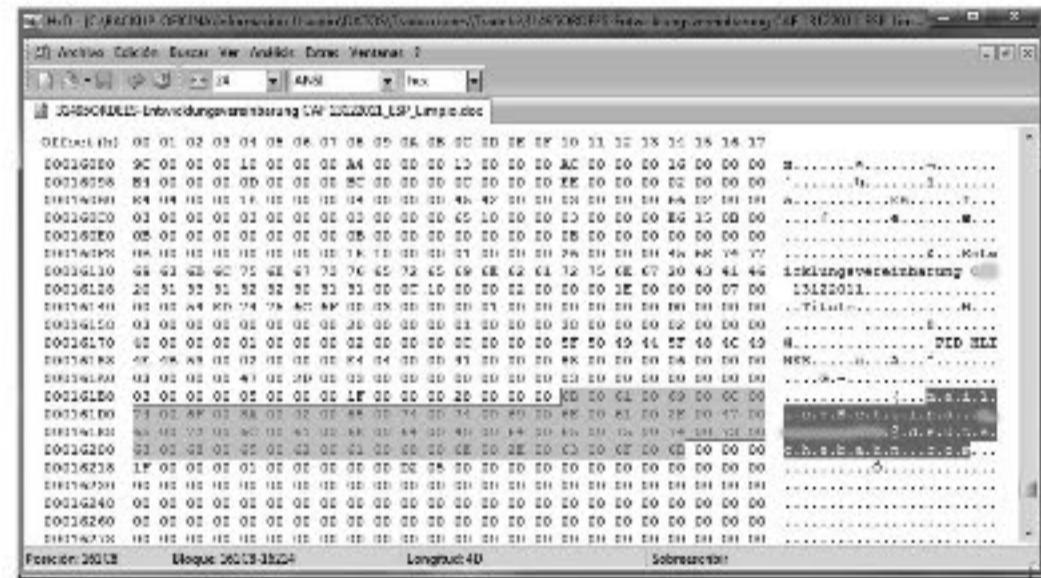


Figura 4.24. Información confidencial de un documento MS-Word vista a través de HxD

#### 4.10.2 Metadata Assistant

Metadata Assistant es un software desarrollado por Payne Consulting (<http://www.payneconsulting.com>), empresa especializada en gestión y eliminación de metadatos en documentos Word para despachos de abogados. Se trata de una herramienta que permite llevar un control de la información oculta en archivos creados por Microsoft Office. Aunque se trata de un software diseñado principalmente para eliminar metadatos e impedir que puedan causar problemas al ser hallados en documentos corporativos o correspondencia entre consultoras, despachos de abogados, clientes particulares y entidades de la administración pública, sus capacidades de detección resultan útiles para el análisis de documentos ofimáticos. La versión 3 de Metadada Assistant ha sido rediseñada para Office 2007 con un mayor rendimiento, un nuevo interfaz de usuario y características avanzadas como la capacidad para detectar metadatos en el interior de archivos comprimidos ZIP.

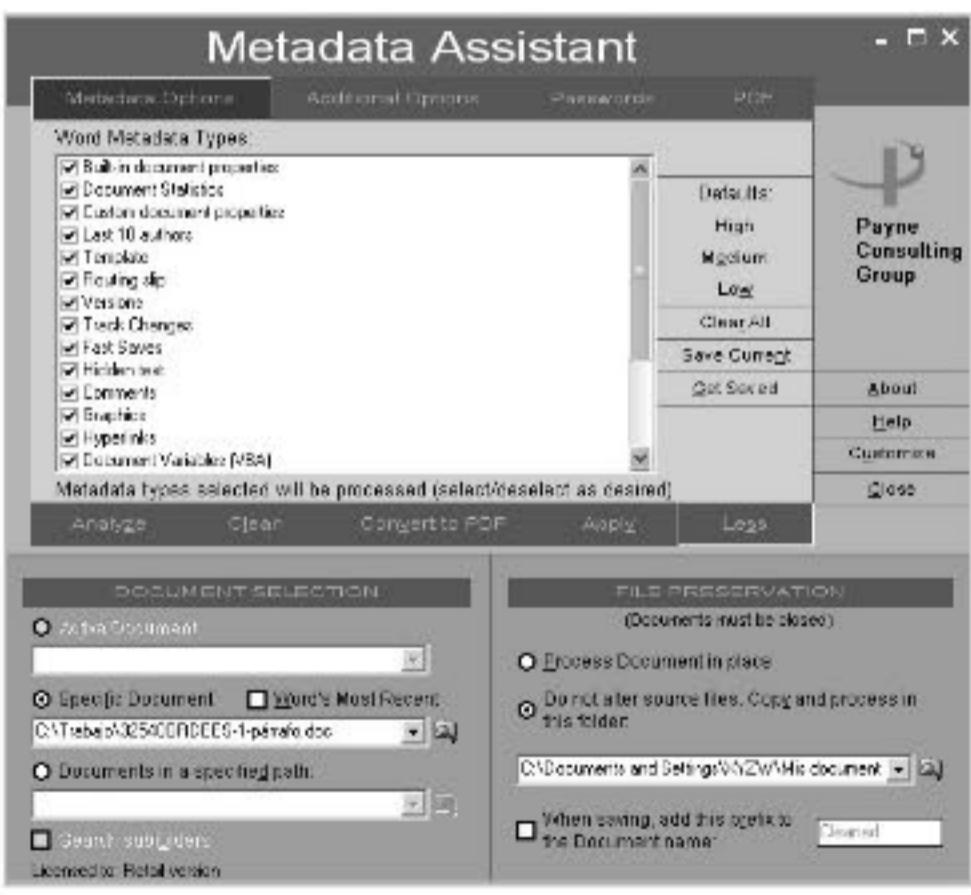


Figura 4.25. Metadata Assistant

Metadata Assistant trabaja tanto con documentos individuales como con lotes de ellos (carpetas). Se comercializa en dos modalidades, una individual y otra de empresa con licencia de instalación para veinte estaciones de trabajo, asistencia técnica y servicio de parches periódicos. El programa posee funcionalidades de gran interés para la investigación forense: integración con Outlook 2007 y Lotus Notes; detección de direcciones de correo electrónico en adjuntos incrustados, archivos ZIP y documentos Adobe PDF, además de la posibilidad de realizar informes de resultados en formato RTF o XML, cuyo propósito originario consiste

en evaluar el nivel de confidencialidad de un documento, pero que el investigador podrá utilizar para obtener protocolos relativos a los elementos de evidencia encontrados y de este modo agregarlos directamente a su informe de investigación.

#### 4.10.3 FOCA

FOCA (*Fingerprinting Organizations with Collected Archives*) es un software desarrollado por la empresa española Informática 64 ([www.informatica64.com](http://www.informatica64.com)) para realizar comprobaciones de seguridad en redes corporativas y sitios web. Obsérvese que esta tampoco es una funcionalidad que tenga que ver con los fines estrictos de una investigación forense. Más bien lo contrario: FOCA se emplea para recopilar información que permita el despliegue de actividades de *hacking* ético y la realización de pruebas de penetración en entornos de redes corporativas con el propósito de evaluar sus condiciones de seguridad y su resistencia ante ataques procedentes del exterior. Pero al igual que en el caso de Metadata Assistant, su capacidad para localizar metadatos en documentos de diversos formatos hace de ella una herramienta útil para el examen de elementos de evidencia. FOCA localiza todos los documentos susceptibles de contener metadatos en un dominio; los descarga al disco duro y después de analizarlos metódicamente presenta los resultados de modo estructurado.

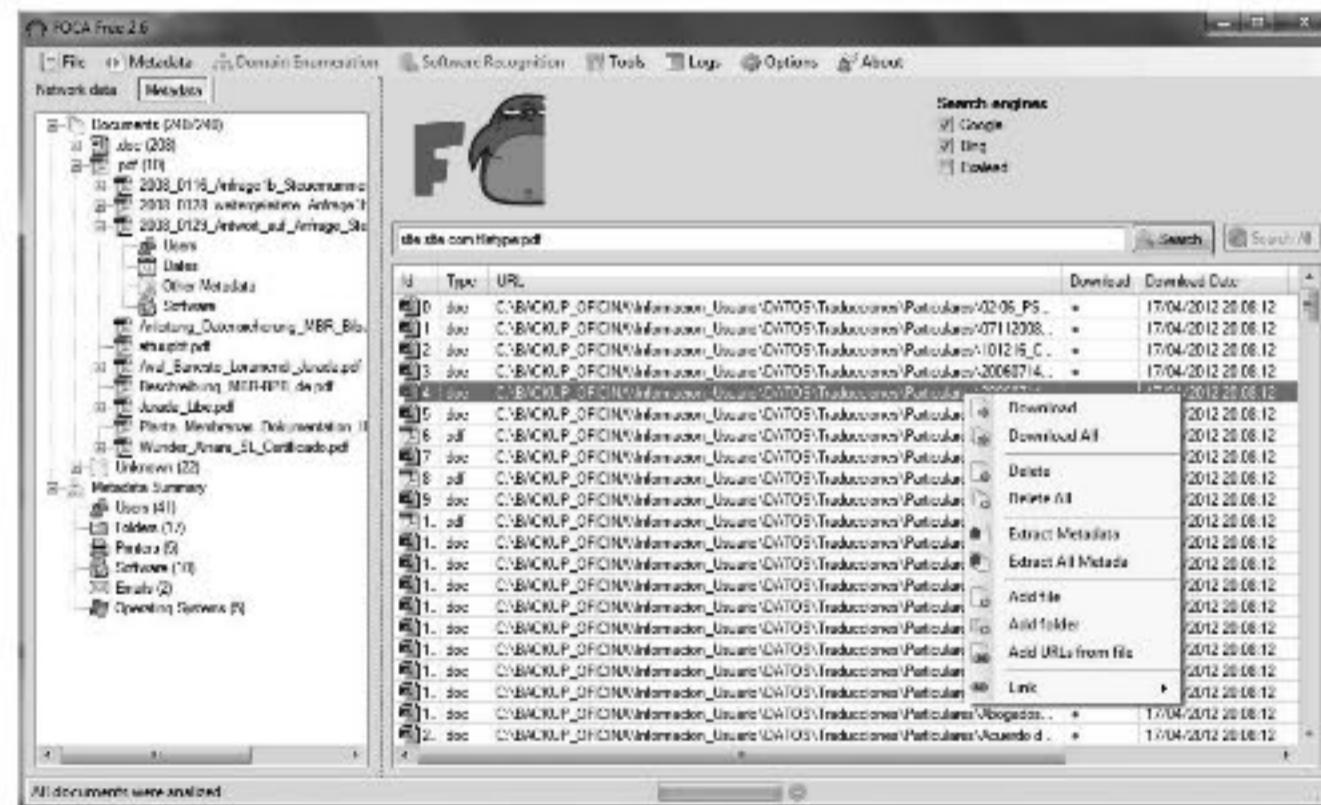


Figura 4.26. FOCA analizando una carpeta de documentos

FOCA está diseñada para mapear redes a partir de la información existente en documentos descargables. Sin embargo la herramienta también puede ser dirigida contra los documentos extraídos de un ordenador sospechoso para obtener elementos de evidencia digital (figura 4.26). FOCA es capaz de rastrear una

enorme variedad de metadatos. A modo de ejemplo he aquí tan solo una breve relación: direcciones de correo electrónico, nombres de usuario, versiones del software utilizado para crear los documentos, servidores internos, información sobre cuotas, carpetas de origen, impresoras, etc. Todo ello desglosado por categorías y de manera perfectamente trazable hasta los documentos de procedencia de los metadatos, que pueden ser examinados en busca de información contextual con un simple clic.

FOCA dispone de una versión de evaluación gratuita. También existe una herramienta *on line* para analizar documentos con metadatos: <http://www.informatica64.com/foca/>. Además de archivos Office, FOCA examina otros formatos como OpenOffice, Corel Word Perfect, Acrobat PDF y JPG.

#### 4.10.4 Metadatos EXIF

Así mismo poseen interés forense los metadatos de archivos gráficos (JPG, PNG, Photoshop PSD), cada vez más abundantes debido a su popularidad en redes sociales así como al uso generalizado de cámaras digitales y software de retoque fotográfico. Los metadatos gráficos existen en una considerable variedad de formatos, como por ejemplo Exif (iniciales de *EXchangeable File Format*) para archivos JPG. También pueden estar formados por etiquetas XML con información cuyo objetivo original consistía en facilitar tareas de administración y localización: fecha y hora, modelo de cámara, longitud focal, velocidad de disparo del obturador, software de retoque utilizado y, en el caso de las etiquetas XML, incluso un historial completo de las operaciones de manipulación llevadas a cabo con Photoshop u otras herramientas de retoque fotográfico. Toda esta información puede servir para extraer elementos de evidencia y ponerlos en el contexto de la investigación. Si la imagen fue tomada con un *smartphone* o teléfono de gama alta equipado con GPS y las opciones de localización del dispositivo están activadas, los metadatos Exif también mostrarán las coordenadas geográficas en el momento de disparar la foto.

Los metadatos Exif pueden visualizarse desde el propio navegador de archivos de Windows. La mayor parte de los visores y herramientas de retoque fotográfico muestran información Exif en el apartado de propiedades de imagen. El investigador también puede utilizar herramientas específicamente diseñadas para la recuperación y gestión de metadatos Exif. Una de las más versátiles y completas es Exiftool de Phil Harvey. Se trata de una utilidad gratuita descargable de <http://www.sno.phy.queensu.ca/~phil/exiftool/>. Exiftool es multiplataforma y en la página del desarrollador se ofrecen, además del código fuente, versiones ejecutables para diversos sistemas operativos.

La versión para Windows permite arrastrar y soltar archivos y carpetas. Si el usuario lo desea también puede renombrar el archivo de aplicación “exiftool(-k).exe” a “exiftool.exe” para utilizarlo como herramienta en línea de comando. La mayor ventaja de Exiftool es que además de metadatos Exif reconoce otros formatos para archivos gráficos (GPS, IPTC, XMP, JFIF, GeoTIFF, ICC Profile, Photoshop IRB, FlashPix, AFCP e ID3) así como etiquetas propietarias correspondientes a cámaras fotográficas de diversas marcas (Canon, Casio, FujiFilm, GE, HP, JVC/Victor, Kodak, Minolta/Konica-Minolta, Nikon, Olympus/Epson, Panasonic/Leica, Pentax/Asahi, Reonyx, Ricoh, Samsung, Sanyo, Sigma/Foveon y Sony). Con Exiftool podrá investigar así mismo metadatos de documentos MS-Office y etiquetas de archivos de sonido MP3. Volveremos a hablar de esta herramienta en el capítulo dedicado a la investigación forense de imágenes digitales.

#### 4.11 ANÁLISIS DE PARTICIONES NTFS Y FAT

Un buen análisis forense no se limita a la obtención de evidencia por medios automatizados. A menudo es necesario trabajar a bajo nivel para recuperar archivos o llegar directamente a los lugares del disco duro que nos interesan. El conocimiento detallado del sistema de archivos reviste un gran interés para la investigación forense. Por lo general los sistemas operativos no permiten el acceso a estructuras de bajo nivel, ni siquiera con permisos de administrador o superusuario. Para sortear este obstáculo existen en el mercado algunas herramientas especiales.

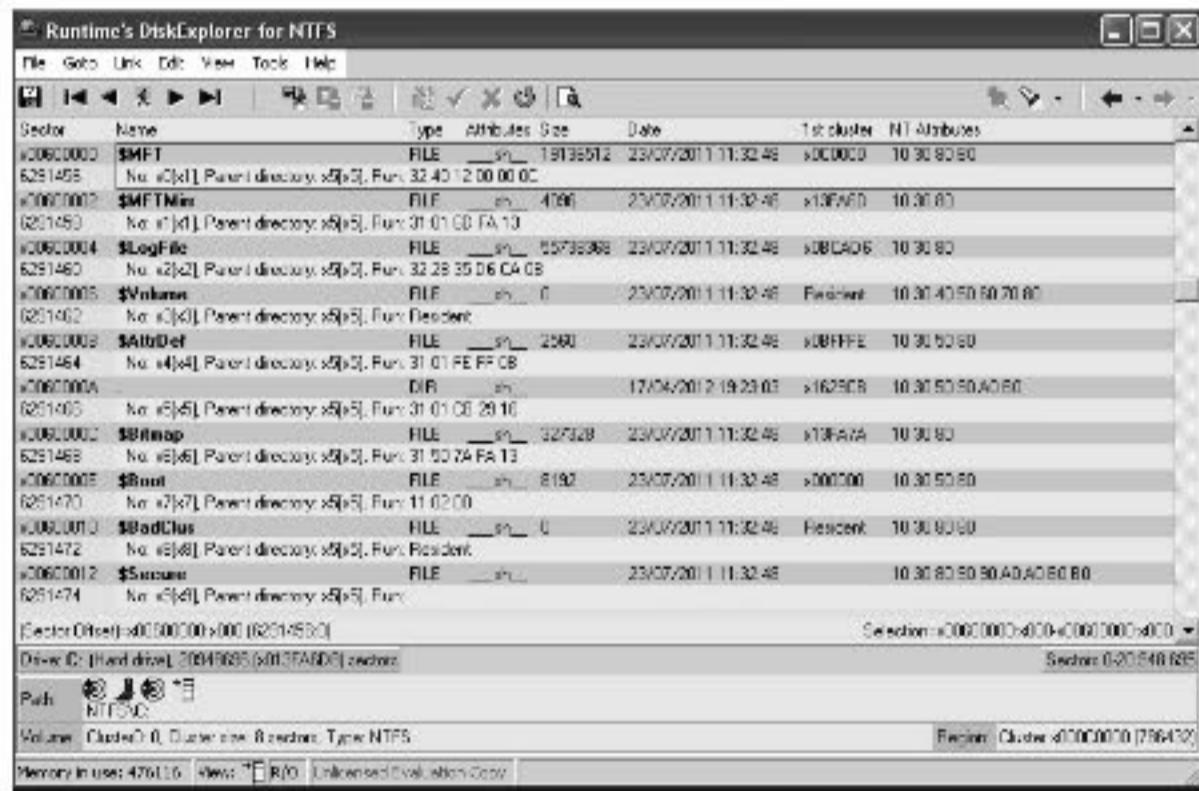


Figura 4.27. Runtime DiskExplorer analizando la Mft de una partición NTFS

### 4.11.1 Runtime DiskExplorer

Runtime es una empresa especializada en desarrollo y comercialización de software para análisis de soportes de datos. Aunque su objetivo consiste en la asistencia técnica en cometidos de recuperación de datos, las herramientas de Runtime pueden prestar también un servicio bastante apreciable al investigador forense. Su producto DiskExplorer, en versiones separadas para FAT y NTFS (figura 4.27), es un editor que permite acceder directamente a elementos críticos del sistema como tabla de particiones, registro de arranque, FAT o Mft (en la versión para NTFS), directorio raíz, e incluso a sectores y *clusters* determinados. Se puede elegir entre varios modos de visualización: hexadecimal, texto, directorio, FAT, tabla de partición y registro de arranque.

DiskExplorer ofrece funciones de búsqueda de caracteres de texto en el disco, en los registros de arranque, tablas de partición y subdirectorios. También muestra detalles del registro de arranque de las particiones. En la versión para NTFS se dispone de información completa sobre cada uno de los registros de la Mft con todos los atributos del sistema de archivos NTFS. La relación de *data runs* hace posible la recuperación manual de archivos borrados.

### 4.11.2 Recuperación de archivos borrados

Los ciberdelincuentes de nuestros días no nacieron ayer. A lo largo de los años, gracias a la experiencia y posiblemente guiados también por las noticias relativas a los casos de investigación más sonados, van adquiriendo un grado cada vez mayor de conocimientos informáticos. Al menos saben lo mínimo para evitar verse atrapados con las manos en la masa. Periódicamente vacían la Papelera de Windows para eliminar pruebas, encriptan sus archivos y se sirven de distribuciones autoarrancables de Linux para acceder a archivos y robar información sin dejar rastros de haber tocado el ordenador. Quizás tengan también alguna herramienta de borrado seguro en su arsenal.

Por fortuna para el investigador la complejidad de los sistemas informáticos juega a favor de la justicia. Que una aplicación Windows cree archivos en un grupo de *clusters* determinado y lo mantenga confinado en él no es lo más normal. Por razones técnicas a las que no podemos dedicar espacio en esta obra, el sistema prefiere dejarlos distribuidos por el disco duro, poniéndolos en carpetas temporales, volcando sus datos al archivo de paginación o colocándolos en los lugares menos previsibles. El cifrado de archivos con aplicaciones como GPG agrava el problema de la dispersión, ya que para trabajar con un archivo cifrado antes es necesario desencriptarlo. Esto supone grabar una versión en texto claro del archivo en otra ubicación del disco duro. Una vez hechos los cambios por el

usuario, el archivo se encripta de nuevo y se borra la versión en texto claro. Y así sucesivamente.

El problema está en que, por lo general, cada vez que un archivo se desencripta, su versión sin cifrar no se graba en los mismos *clusters* del disco duro que ocupó la vez anterior, sino en cualquier otra parte del medio. Es el sistema quien lo decide, sin que el usuario tenga la menor capacidad para influir sobre ello. De este modo el disco duro va quedando salpicado de versiones borradas del documento en texto claro, que se podrán recuperar con ayuda de alguna de las numerosas aplicaciones de recuperación de archivos borrados existentes en el mercado.

No podemos recomendar al lector una herramienta concreta. Lo más aconsejable es que él mismo busque la que se adapte de manera óptima a sus necesidades o a los requisitos de su empresa. La mayor parte de estas herramientas son de manejo fácil e intuitivo, y aunque sean de pago no tienen un coste prohibitivo. El lector podrá probarlas y decidir por sí mismo, ya que casi todas disponen de versiones demo con una funcionalidad limitada —por lo general permiten visualizar los archivos recuperables detectados sin permitir la copia de los mismos a una unidad de destino—. De igual modo se pueden utilizar las técnicas aprendidas en el capítulo 3 para recuperar archivos con The Sleuth Kit. EnCase y FTK también recuperan datos borrados, tanto a partir de la información existente en las estructuras del sistema de archivos como mediante procesos de *data carving*. Lo que sigue, más que como una recomendación, ha de interpretarse en términos de preferencias personales del autor.

Siempre que se utilice una herramienta de recuperación, los archivos salvados deberán guardarse en una partición o en un medio de destino distinto a aquel del cual son extraídos. Y ello por una razón obvia: incluso en operaciones simplemente técnicas, en las que no haya que proteger evidencias o cadenas de custodia, no nos interesa que los archivos recuperados se escriban sobre otros datos que también podrían rescatarse.

### 4.11.3 Runtime GetDataBack

GetDataBack es una utilidad desarrollada por los creadores de la herramienta DiskExplorer a la que ya se ha hecho mención en el apartado 4.11. Se trata de un software automatizado que mediante un asistente inicial permite definir los parámetros de la operación de búsqueda y rescate de datos. El programa existe en dos versiones, una para FAT y otra para NTFS (figura 4.28). Funciona rastreando registros de la Mft —o entradas de directorios en el caso de la versión FAT— y realiza un inventario completo de todos los archivos, incluyendo los

borrados. Sus funcionalidades no incluyen la de recuperar archivos en función de su contenido, pero sí permite seleccionarlos por tamaños y fechas.

En <http://runtime.org/data-recovery-software.htm> podrá encontrar una versión demo de este software. Recuerde que existe un programa para particiones NTFS y otro para particiones FAT. Aunque en la práctica GetDataBack for NTFS se puede utilizar para recuperar también archivos de particiones FAT, no existen garantías de que la operación pueda llevarse a cabo con el mismo grado de eficacia que mediante el empleo de la herramienta específica.

La versión demo muestra los archivos pero no permite recuperarlos. Conviene recordar que GetDataBack no es un asistente de reparación al estilo de las utilidades Norton o Chkdsk. Con él no se pueden reparar particiones ni estructuras de datos. Lo único que hace es recuperar archivos. Esto, por otro lado, lo convierte en una herramienta extremadamente útil en caso de fallos en el sistema o un intento de borrado por parte del usuario sospechoso.

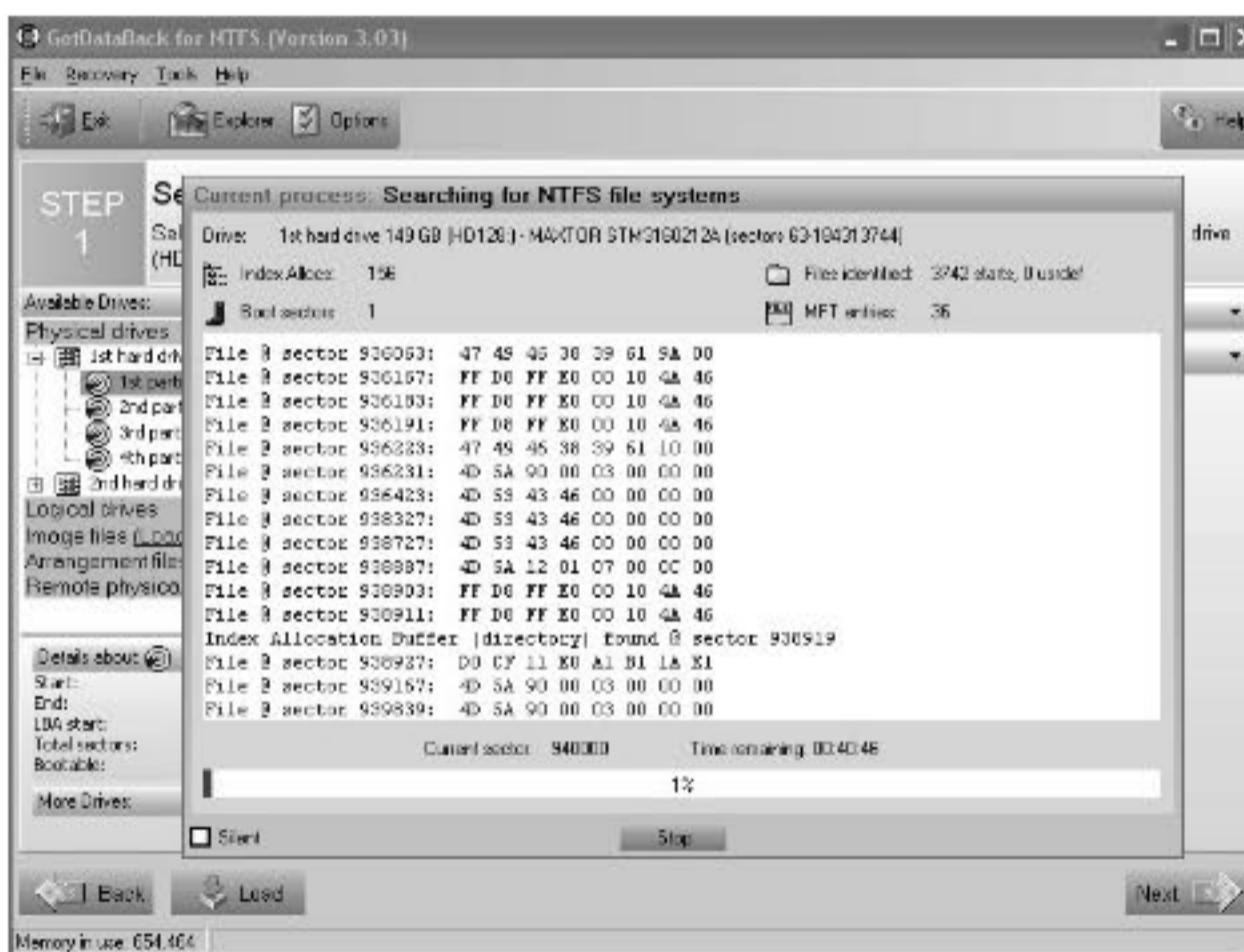


Figura 4.28. GetDataBack for NTFS explorando un soporte de datos

#### 4.11.4 EasyRecovery Professional

EasyRecovery ha sido creada por Ontrack Data, empresa que tiene cierta reputación en el campo de la recuperación de datos. Se trata de una herramienta de manejo sencillo e intuitivo con un modo de funcionamiento automático que trabaja

sin muchas posibilidades de configuración. Carece de modo manual, pero resulta eficaz para recuperar una partición que ha sido formateada recientemente sin escribir después nada encima.

#### 4.11.5 R-Studio

R-Studio (<http://www.r-studio.com>) es una herramienta creada para la recuperación de archivos mediante la aplicación de una metodología creada por la empresa desarrolladora: IntelligentScan. IntelligentScan consiste en llevar a cabo un escaneado minucioso del medio de datos en el transcurso del cual el programa lee los datos directamente del disco, y después de analizarlos intenta determinar a qué estructuras concretas del sistema de archivos se hallan asociados. R-Studio reconoce numerosos tipos de registro correspondientes a los sistemas de archivos de uso más extendido (NTFS, FAT, ext2/3/4 y HFS/HFS+).

Una característica interesante de este software es que a la hora de reconstruir particiones no se decide por una de las posibilidades descartando las demás, sino que ofrece varias alternativas, cada una con su estructura de directorios y sus listados de archivos correspondientes. Así mismo admite la posibilidad de llevar a cabo correcciones manuales en los parámetros de las particiones en caso de descubrirse información adicional sobre las mismas. R-Studio oferta distintas modalidades de licencia según el número de estaciones de trabajo en las que se desee instalar este software de recuperación de datos.

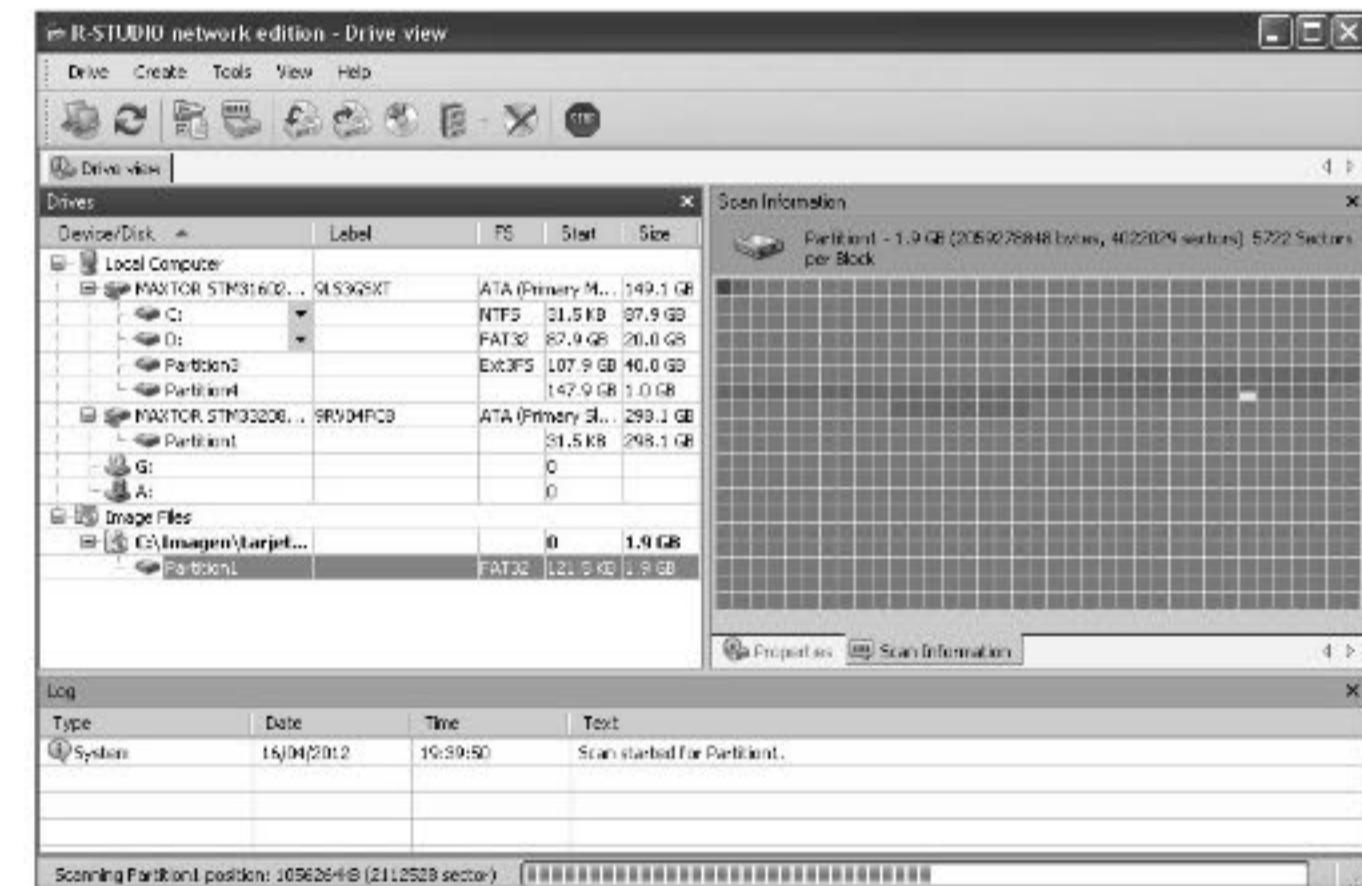


Figura 4.29. R-Studio

## 4.12 EL REGISTRO DE WINDOWS

El hecho de dedicar tan poco espacio al Registro de Windows y haberlo dejado para el final no debe inducir a una impresión equivocada acerca de la importancia que tiene este apartado de la investigación forense en entornos MS-Windows. Seguramente el lector habrá oido hablar de esa compleja base de datos en la que Windows guarda todos los detalles relativos a su funcionamiento y configuración, desde las aplicaciones que van asociadas a cada extensión de archivo hasta el más insignificante ajuste en el arranque del sistema, pasando por las aplicaciones instaladas y las claves de registro de estas. La investigación forense del Registro de Windows constituye un amplio campo sobre el que se podrían escribir libros enteros, no solo por su enorme complejidad técnica sino también porque su abundancia de datos lo convierte en un verdadero filón de elementos de evidencia. Analizando el Registro se puede saber si el sospechoso conectó una llave USB para copiar archivos, si en el ordenador había instalada una determinada aplicación que después fue borrada para eliminar pruebas, cuáles son los últimos archivos abiertos por el usuario o si el sistema está contaminado por troyanos o *malware*.

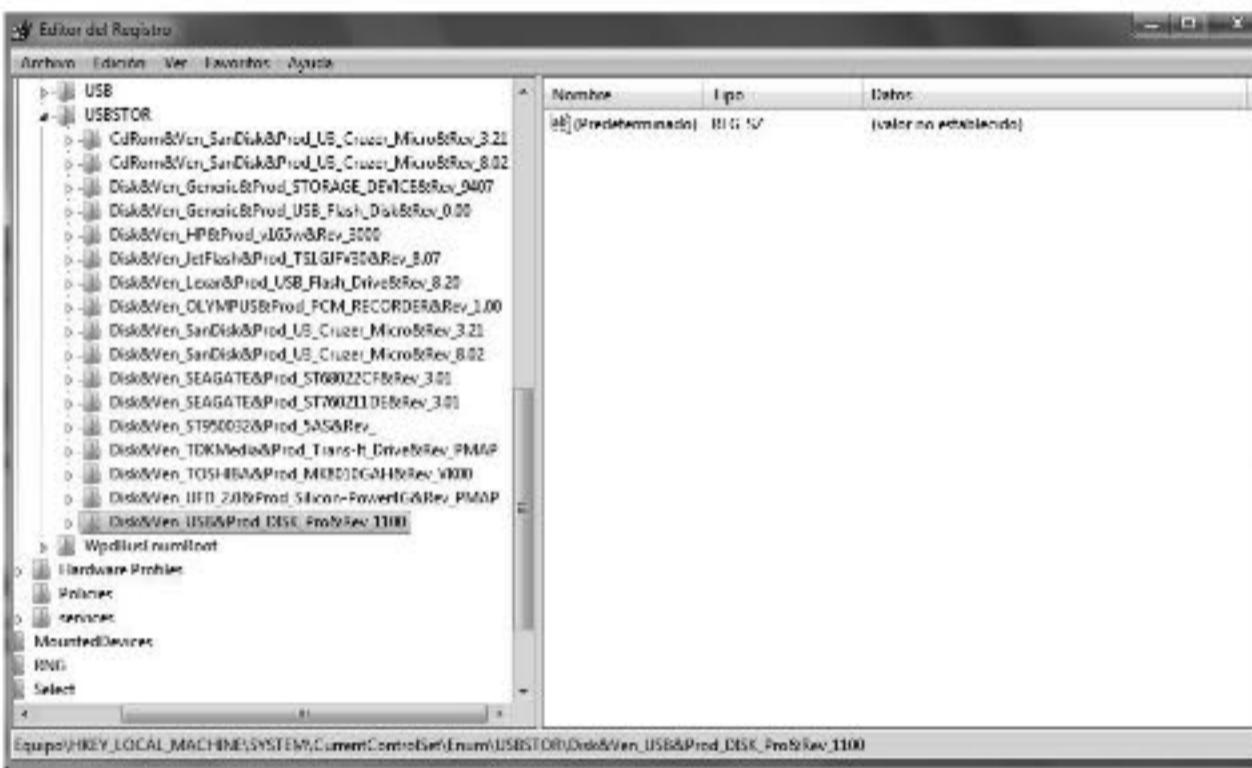


Figura 4.30. Evidencia de soportes USB en el Registro de Windows

Por desgracia no podemos extendernos ni siquiera a los planteamientos esenciales del análisis del Registro de Windows. En las páginas que siguen únicamente se proporciona al lector una introducción al funcionamiento del Registro en aspectos básicos relacionados con la investigación forense, junto con algunos ejemplos de la información que se puede conseguir a través de algunas claves determinadas en situaciones típicas. Con ello confío en poder transmitir una idea adecuada del método de trabajo seguido en relación con el Registro y motivar

al lector para que siga adelante por sus propios medios. Moverse por el Registro no es fácil. Aunque su estructura es simple, la ramificación de claves, subclaves y valores de todo tipo, sin posibilidad de sistematización por medio de una teoría asimilable en unos pocos principios básicos, hacen que —al margen de los peligros que representa cualquier manipulación no autorizada del Registro— la exploración de esta parte de los sistemas operativos de Microsoft, más que a la visita programada de una fábrica con unas cuantas máquinas y relaciones de funcionalidad perceptibles a simple vista, se asemeje a un paseo a través de un jardín botánico poblado por millares de especies de las cuales el usuario ni siquiera conoce el nombre.

Para profundizar en el tema puede recurrir a las obras citadas en la bibliografía, particularmente al libro de Harlan Carvey sobre *Análisis Forense de Sistemas Windows*, obra imprescindible de referencia en la materia en la que hay todo un capítulo dedicado al Registro.

### 4.12.1 Estructura y archivos del Registro

El Registro es una estructura de datos que se forma durante el arranque del sistema a partir del contenido de varios archivos que en el caso de los sistemas Microsoft de productividad avanzados Windows 2000/XP/Vista/7 se encuentran en dos directorios: (i) %systemroot%\system32\config (con los archivos SAM, Software, Security, Default y System) y (ii) la carpeta de perfil de usuario (archivo NTUSER.DAT). Una vez desplegada toda esta información tras el inicio del sistema, se construye de manera automática un grupo lógico de claves y subclaves con sus valores correspondientes a las que Microsoft denomina Registry Hives, y que permanecerán activas hasta que el sistema operativo se detenga o el ordenador tenga que ser reiniciado por causa de un problema técnico. En Windows 7, por ejemplo, el árbol de claves posee cinco ramas principales que se detallan a continuación:

- HKEY\_CLASSES\_ROOT
- HKEY\_CURRENT\_USER
- HKEY\_LOCAL\_MACHINE
- HKEY\_USERS
- HKEY\_CURRENT\_CONFIG

Los archivos que conforman la base del Registro son actualizados de manera permanente durante el funcionamiento del sistema, por ejemplo cada vez que el usuario inicia sesión o conecta una llave USB al ordenador (figura 4.30). Resulta posible examinar toda esta estructura de datos a través de un editor incluido en el sistema operativo al que no podrá acceder a través del menú de Inicio ni de ningún ícono. Existe para ello una buena razón: al personal de servicio al cliente de Microsoft le interesa que el usuario utilice su ordenador para tareas de productividad y no para hacer experimentos con una compleja base de datos cuya más mínima perturbación podría dejar inutilizado el sistema operativo.

Para llegar hasta el editor del Registro es necesario llamarlo explícitamente desde el menú **Ejecución de programa** (en el botón de **Inicio**) o abriendo una consola de texto:

```
C:\Users\Usuario>regedit.exe
```

Una vez desplegado el Registro podrá navegar por él examinando claves, subclaves y valores. Si se trata de obtener evidencia en el contexto de una investigación con el sistema en funcionamiento no hay peligro, siempre que el investigador tenga cuidado en no introducir el más mínimo cambio. En los numerosos apartados del Registro podrá observar infinidad de datos correspondientes a la configuración del ordenador y la actividad del usuario. En la figura que se acompaña, por ejemplo, se puede apreciar una lista de los dispositivos USB que en algún momento han sido conectados al ordenador, con identificadores únicos de cada fabricante, modelos y códigos de revisión de hardware. Si el investigador encuentra algún elemento de evidencia que pueda ser útil para su caso, lo podrá extraer exportando el contenido de la clave en forma de archivo con extensión REG.

## 4.12.2 Análisis off line con Windows Registry Recovery

Regedit.exe no es la única herramienta de la que dispone el investigador para examinar el contenido del Registro. Este también puede ser analizado con el ordenador apagado, utilizando para ello herramientas de terceros que trabajan sobre la imagen adquirida del soporte de datos o directamente sobre los propios archivos en los cuales se guarda la información correspondiente al Registro de Windows. Esto ofrece la ventaja de que en caso de que en el sistema hubiera instalado algún *malware* o *rootkit*, este no podrá ocultar su presencia alterando el flujo de datos en vivo. Hasta hace relativamente poco el análisis *off line* del Registro no era algo habitual, debido al desconocimiento de su estructura interna y la falta de una metodología de aceptación general. En la actualidad, sin embargo, se dispone de herramientas eficaces para ello.

Una de las más recomendadas por los expertos es Windows Registry Recovery de la empresa Mitec. A partir de los propios archivos del Registro esta aplicación permite obtener elementos de evidencia de gran interés para una investigación forense, como por ejemplo:

- Información general del sistema.
- Cuentas de usuario.
- Software instalado y parches.
- Servicios del sistema.
- Etc.

## 4.12.3 RegRipper

RegRipper es un *script* en Perl escrito por Harlan Carvey cuya función consiste no en proporcionar al usuario un interfaz para navegar entre las diferentes claves, sino más bien en interpretar los contenidos de aquellas de acuerdo con determinados patrones de búsqueda. Mediante la aplicación de una batería de *plugins*, RegRipper extrae información y la introduce en un archivo de informe que después el investigador puede examinar a su conveniencia.

Esta herramienta es gratuita y se halla a disposición del público, pudiendo ser descargada desde la dirección <http://www.regrripper.net>, donde el investigador hallará además gran cantidad de documentación y recursos sobre la misma, así como un foro especializado.

\* \* \*

Todas las marcas, denominaciones de software, diseños de interfaces, ilustraciones, esquemas y conceptos de ingeniería de carácter comercial a los que se ha hecho mención en el presente capítulo son propiedad intelectual de las empresas que se mencionan en los apartados respectivos.

## ANÁLISIS FORENSE DE SISTEMAS LINUX/UNIX

### Capítulo 5

Existen numerosas razones para tratar con cierto grado de detalle la investigación forense de sistemas Linux/Unix. En primer lugar Linux y el software libre son fenómenos tecnológicos y culturales de gran importancia ligados a la popularización de la informática y el despegue de Internet durante los años 90 del siglo pasado y la primera década del presente. Algunas de las herramientas que se han estudiado en capítulos anteriores fueron diseñadas en un principio para entornos Unix y adaptadas con posterioridad a otras plataformas. Conviene que el investigador posea nociones de unos sistemas operativos y aplicaciones de software que le pueden servir para montar estaciones de trabajo sin la cuantiosa inversión que requiere un sistema comercial como EnCase o FTK. Finalmente es un hecho conocido que durante los últimos años la proliferación de dispositivos móviles –agendas electrónicas, *smartphones*, libros electrónicos, reproductores de medios, etc.– ha traído consigo la presencia en el mercado de otros sistemas operativos basados en Linux/Unix que se apartan de la filosofía de diseño y la estética habitual del entorno MS-Windows, como por ejemplo Android o Apple OSX. Un conocimiento de Linux resulta imprescindible para moverse en numerosos ámbitos de la investigación forense actual.

### 5.1 HERRAMIENTAS DE CÓDIGO LIBRE

#### 5.1.1 ¿Qué es exactamente el código libre?

Sin ser este lugar para una explicación fundamentada acerca de lo que es el código libre, o sobre su filosofía, orígenes y particularidades de los más de cincuenta tipos disponibles de licencia de la *Free Software Foundation*, hay tres cosas importantes que resumen la esencia de todo el tema y que el investigador debe saber. Se dice que un software es libre –lo cual no implica que tenga que ser también gratuito– si cumple las condiciones siguientes:

- a) Su desarrollador ha establecido que tanto los archivos ejecutables del programa como el código fuente del mismo estén a disposición del público.
- b) El usuario podrá descargarse el programa y utilizarlo sin restricciones. También podrá examinar el código fuente y realizar cambios en él con el objeto de adaptarlo a sus necesidades o incluir características de funcionamiento que le sean útiles.
- c) No obstante, y si después de haber introducido modificaciones en el programa el usuario decide ponerlo a disposición del público, estará obligado a difundir no solamente los archivos ejecutables sino también el código fuente con los cambios realizados en las mismas condiciones en que el autor originario del software puso a disposición el suyo.

No nos vamos a ocupar aquí del debate en torno al software libre ni de sus contribuciones a la Sociedad de la Información. Para los propósitos que aquí se persiguen basta saber que se trata de un tipo especial de software utilizado en el desarrollo de herramientas que, además de potencia, fiabilidad, bajo coste y haber sido creadas por expertos de acreditado renombre en el campo de la investigación forense, en términos legales ofrecen también algunas ventajas. El estar reconocidas por diversas autoridades e instituciones y la disponibilidad del código fuente dificultan en un juicio los argumentos basados en una posible alteración del programa o la falta de transparencia en el funcionamiento de aplicaciones forenses.

#### 5.1.2 Linux en la investigación forense

La gran ventaja de las herramientas forenses comerciales como EnCase o FTK es que ahorran trabajo al investigador. No es lo mismo poder rescatar cientos o miles de archivos borrados con unos pocos clics de ratón que tener que ir a por ellos tecleando órdenes en línea de comando o a base de *scripts* que permitan

automatizar el funcionamiento de TSK. El elevado coste de las herramientas comerciales no suele ser problema para las entidades que las emplean, por lo general fuerzas del orden público o departamentos de seguridad de las grandes empresas. Sin embargo existen situaciones en las que la utilización de Linux y herramientas de código libre supone beneficios no solo de coste, sino también operativos: realización de imágenes *in situ* cuando no se dispone del material utilizado en el puesto de trabajo, tareas específicas de comprobación y análisis, recuperación de archivos borrados mediante técnicas de *data carving*, etc.

Otro de los beneficios de Linux y el código libre reside en el hecho de ofrecer nuevas oportunidades para el aprendizaje de la profesión. Hace años las dos únicas vías curriculares para el investigador forense eran la policía y las fuerzas armadas. Actualmente el código libre proporciona tecnología, herramientas, recursos y documentación abundante a todo aquel que se interese por la investigación forense, y también a centros universitarios y de formación profesional para que puedan organizar programas docentes y actividades relacionadas con la investigación de medios digitales aprovechando la infraestructura de equipos y redes existente y sin tener que realizar cuantiosas inversiones en personal especializado y software.

### 5.1.3 Poniendo en marcha una estación de trabajo con Linux

La investigación forense de sistemas informáticos basados en Linux/Unix puede llevarse a cabo sin problemas mediante herramientas comerciales como EnCase o FTK. Las razones por las que decidimos utilizar una plataforma Linux son más que nada de afinidad funcional. El mismo sistema de archivos, árboles de directorios similares, archivos de registro muy parecidos y la misma filosofía de diseño y funcionamiento. No hace falta mencionar que Linux como plataforma forense no solo sirve para analizar sistemas basados en Linux, sino también en Microsoft Windows como se ha visto en el capítulo anterior, y en general cualquier plataforma que esté basada en la arquitectura X86.

El investigador dispone de una amplia gama de distribuciones. No tiene más que darse una vuelta por la página web [distrowatch.com](http://distrowatch.com) y elegir la que más le convenga. Si se decide por Ubuntu o alguno de sus derivados tendrá la ventaja de que la mayor parte de las herramientas que va a utilizar vienen ya incluidas en la distribución o están disponibles en los repositorios oficiales. La instalación de paquetes en Ubuntu se lleva a cabo mediante un interfaz gráfico (Aptitude, Adept) o en línea de comando tecleando:

```
sudo apt-get install [nombre del paquete]
```

### 5.1.4 Descarga, compilación e instalación de herramientas

En caso de que tenga previsto incorporar características nuevas o desee disponer de la versión más reciente de una herramienta, podrá descargar su código fuente desde la página web del desarrollador, compilarla e instalarla por el procedimiento habitual. Normalmente el código fuente se obtiene comprimido en un archivo tipo tarball con extensiones .tar.gz o tgz. Para desempaquetarlo es preciso tener instaladas todas las utilidades que permiten gestionar archivos comprimidos, y también las herramientas de desarrollo: compiladores de C++, make, intérpretes de Perl, Python, Java, etc. —en el supuesto de que la herramienta forense los necesite para funcionar— y todas las librerías correspondientes.

Veámoslo con un ejemplo muy similar al que ya se expuso en un capítulo anterior para la instalación de TSK. Supongamos que queremos instalar una herramienta llamada Exiftool, a la que ya se ha hecho mención en el apartado correspondiente a la investigación de metadatos gráficos y de la cual volveremos a tratar más adelante, que sirve para recuperar y analizar metadatos de archivos gráficos. En primer lugar obtenemos la herramienta bajándola desde la página web del desarrollador:

```
wget http://www.sno.phy.queensu.ca/~phil/exiftool/Image-ExifTool-X.XX.tar.gz
```

Sustitúyanse las “XX” por los números de versión. El resto es rutina para el usuario experimentado de Linux:

```
gunzip < Image-ExifTool-X.XX.tar.gz | tar xvf -
cd Image-ExifTool-X.XX
perl Makefile.PL
make test
make install
```

En cualquier caso deberá consultarse la documentación de acompañamiento del código, por lo general en forma de archivos del tipo README, doc, rtf y demás.

### 5.1.5 Montaje automático de particiones

Como ya se ha indicado en más de una ocasión, el aspecto más delicado de la investigación forense tiene que ver con la integridad de la evidencia y la conservación de la cadena de custodia. El más mínimo cambio en los archivos adquiridos —detectado mediante el empleo de programas que calculan las sumas de verificación— puede traer consigo una impugnación de la evidencia por la parte contraria en un proceso. El montaje automático de particiones con *journaling* (por ejemplo cuando se trata de sistemas de archivos NTFS, ext3, ReiserFS) constituye

un caso típico de este problema. Las modificaciones que el proceso de arranque introduce en el archivo de *journaling* bastan para hacer que el *hash* del soporte analizado por el investigador no coincida con el de las copias en poder del tribunal y la parte contraria. El resultado podría ser catastrófico para la defensa de nuestra posición en el caso.

En distribuciones antiguas de Linux el montaje automático de particiones no constituía un problema porque no existía: todo se hacía a mano de manera explícita mediante la creación de directorios como puntos de montaje y el empleo del comando *mount*. Esto se puede seguir haciendo con las distribuciones nuevas. Por ejemplo, si tenemos un disco duro PATA procedente de la incautación policial de un ordenador con Windows XP y una partición con sistema de archivos NTFS, el procedimiento para acceder a sus contenidos desde Linux sería más o menos el siguiente. En primer lugar conectamos el soporte al segundo interfaz ATA paralelo después de haber configurado el *jumper* de manera conveniente. A continuación arrancamos el sistema (supongamos que se trata de una distribución Linux con el montaje automático de particiones desactivado). Desde una consola de texto *bash* y con privilegios de administrador creamos un directorio que sirva como punto de montaje:

```
mkdir discoXP
```

Después montamos la unidad y accedemos a ella como si se tratara de un árbol de directorios normal. También podemos utilizar un navegador de archivos:

```
sudo mount /dev/sda1 -t ntfs ./discoXP
cd discoXP
...etc
```

Las distribuciones populares basadas en Ubuntu y Fedora ofrecen una amplia funcionalidad de reconocimiento de hardware y montaje automático de particiones. Esto lo consiguen gracias a la interacción de diversos módulos de software que al iniciarse el sistema arrancan en forma de daemons: *udev*, cuya misión consiste en crear todo el sistema de nodos para la gestión de dispositivos; *HAL* (*Hardware Abstraction Layer*, Capa de Abstracción de Hardware), el cual mantiene en memoria información sobre dispositivos conectados actuando como intermediario en todo proceso de comunicación con los mismos, y *d-messagingbus* o bus de mensajes del sistema, mecanismo que permite a las aplicaciones entrar en comunicación e intercambiar datos unas con otras.

El funcionamiento coordinado de estos tres elementos de software es lo que hace posible que, cada vez que insertamos un *pendrive* o un disco duro externo en cualquiera de los conectores USB del ordenador, en nuestro escritorio GNOME

o KDE aparezca automáticamente un ícono con la opción para abrir un navegador de archivos. Pero esto, que tan práctico resulta para la mayor parte de los usuarios, puede dar al traste con el trabajo de un investigador que solo quiera hacer copias a bajo nivel de discos duros u otros soportes en los que existan particiones con *journaling*. Siendo ya difícil hallar investigadores capacitados para el análisis de sistemas Unix/Linux, lo único que falta es que la tarea del fiscal o del letrado se eche a perder por no haber tomado las debidas precauciones. Se trata por consiguiente de una fase crítica en el tratamiento de la evidencia digital.

Existen tres soluciones posibles al problema del montaje automático de particiones. La primera consiste en desactivar *udev*, *HAL* y *d-messagingbus* mediante cambios en los *scripts* de arranque de Linux. Esta opción, realizable sin muchos problemas en distribuciones como Slackware, que están diseñadas para usuarios experimentados y admiten un alto grado de granularidad en la configuración del sistema, se vuelve poco práctica en Ubuntu, cuya facilidad de manejo depende en gran medida de la actuación inicial de esos módulos de software. Si desactivamos *udev*, *HAL* y *d-bus*, puede que lo que tengamos delante una vez completado el proceso de arranque siga siendo Ubuntu, con sus atractivos y cómodos sistemas de ventanas GNOME o KDE, pero puede apostar a que no se comportará como el Ubuntu que está acostumbrado a utilizar.

La segunda alternativa pasa por el empleo de una distribución de la cual estemos completamente seguros que no monta particiones con *journaling* de modo automático. Ante la ausencia de información fiable acerca del tema el único recurso que queda es probarlo nosotros mismos, a base de comparar el *hash* del soporte originario con el de la imagen adquirida después de arrancar el sistema operativo con el soporte conectado a través del interfaz ATA paralelo, SATA o USB correspondiente. En experimentos realizados por él mismo, el autor de este libro ha podido averiguar que Slackware 13.0 con los *scripts* modificados para impedir el arranque automático de *udev*, *HAL* y *d-bus* no monta particiones con *journaling* del tipo ext3 y ReiserFS, a no ser que el usuario lo ordene de un modo explícito mediante el comando *mount*. SystemRescueCD tampoco lo hace. Knoppix por el contrario sí, lo cual la incapacita como distribución forense a no ser que se utilicen medidas de seguridad como el uso de bloqueadores de escritura.

Esta, la de bloqueadores de escritura, es precisamente la tercera y más práctica solución al problema: vía del hardware. El investigador debería adquirir un juego completo de ellos para los diferentes tipos de interfaz (PATA, SATA, USB) y hacer constar en lugar bien visible del informe que fueron utilizados sin falta.

## 5.2 ESTRUCTURA TÍPICA DE UN SISTEMA LINUX

Linux es un sistema operativo multitarea compuesto por un conjunto de aplicaciones —las cuales no tienen por qué ser todas ellas de código libre— que funcionan sobre un *kernel* o núcleo cuya versión original fue creada en 1991 por el informático finlandés Linus Torvalds. Desde entonces se han sucedido diferentes versiones de este *kernel*, siendo la 3.0 la que se utiliza en la actualidad. La dinámica de desarrollo del *kernel* es comunitaria: un equipo de programadores de todo el mundo coopera a través de Internet en el perfeccionamiento del núcleo, la solución de problemas y el desarrollo de módulos y características nuevas. Las ediciones sucesivas o *releases* del *kernel* se llevan a cabo bajo la licencia GNU GPL, que permite la distribución y la incorporación de modificaciones, pero exige que todas las copias del trabajo original y sus cambios sean puestos a disposición del público en los mismos términos, y que el código fuente siempre pueda ser obtenido en los mismos términos que el software licenciado. La coordinación de los trabajos de programación y su incorporación a las versiones liberadas del *kernel* las lleva a cabo el propio Linus Torvalds. El *kernel* está disponible para descarga en el sitio web <http://www.kernel.org>.

### 5.2.1 Arquitectura y sistemas de archivos

Linux es básicamente un núcleo monolítico compuesto por una pieza principal de software denominada *kernel*. Los controladores de dispositivos y las extensiones del núcleo se ejecutan en un espacio privilegiado conocido como anillo 0, lo cual asegura acceso ilimitado al hardware. Otros elementos de software se ejecutan en el espacio de usuario. Controladores y extensiones se pueden cargar y descargar en forma de módulos, incluso con el sistema operativo en funcionamiento. Estando acostumbrados a la idea de que Windows utiliza exclusivamente particiones NTFS y FAT, cuesta asimilar la noción de algo que es capaz de acceder a una variedad virtualmente ilimitada de sistemas de archivos: ext2, ext3 (actualmente el sistema nativo de Linux) y ext4, ReiserFs, Reiser4, JFS, XFS, NTFS, FAT en todas sus versiones históricas (FAT12, FAT16 y FAT32), minix, HFS y HFS+, etc. Con una estación de trabajo Linux y un *kernel* compilado para incorporar al mismo las opciones y características funcionales precisas, el investigador podrá acceder a cualquier tipo de partición y —con la ayuda de hardware y controladores adecuados— a la práctica totalidad de soportes de datos existentes en el mercado, incluyendo dispositivos antiguos como lectores de cinta, unidades magnetoópticas, etc.

En Linux todo está representado por archivos: no solo ejecutables, datos del usuario y directorios —que no son sino archivos que contienen los nombres de otros archivos—, sino también el hardware, todos los periféricos y los medios de almacenamiento con sus diferentes particiones y volúmenes. Este principio de

abstracción permite organizar un sistema completo dentro de una jerarquía unificada de directorios. Puede haber variantes de unas distribuciones a otras, pero del mismo modo que Linux sigue el esquema general de organización de Unix, solemos hallar la misma jerarquía de directorios no solo en sistemas de sobremesa, sino también en ordenadores Apple OSX (incluyendo iPhone), dispositivos móviles basados en Android e incluso maquinaria y artefactos electrónicos que incluyan algún tipo de software desarrollado a partir de Linux.

En Linux no existen las unidades típicas de Windows representadas por letras (C:, D:, E:, etc.). El acceso a una partición se logra “ensamblándola” dentro del árbol de directorios —del cual hablaremos en el apartado siguiente— a través de un punto de montaje, como ya se ha visto en el ejemplo del apartado 5.1.5. Este no es otra cosa que un directorio dentro del cual aparecen listados los contenidos del soporte después de haberlo montado mediante el comando `mount`. Accediendo a este punto de montaje con la ayuda de los comandos de Linux que permiten cambiar de unos directorios a otros, el usuario podrá moverse en el interior del soporte de datos exactamente del mismo modo que cuando selecciona su unidad característica en un entorno MS-Windows.

### 5.2.2 Jerarquía de directorios

Para garantizar la compatibilidad y portabilidad, tanto con Unix como entre las diversas distribuciones Linux, los sistemas Unix cumplen un denominado FHS (*Filesystem Hierarchy Standard*: Estándar de Jerarquía de Sistemas de Archivos) que especifica una disposición de elementos con el objeto de que programadores, administradores de sistemas y usuarios sepan dónde instalar o buscar los recursos que necesitan. Esta disposición tiene un carácter más riguroso que en Windows —donde a condición de no tocar la carpeta del sistema operativo y algunas ubicaciones protegidas el usuario puede instalar software y guardar datos donde desee—. Aunque este estándar no es aplicado por los desarrolladores al 100% las diferencias entre unas distribuciones y otras suelen ser mínimas.

El punto de montaje principal de un sistema Linux, del cual brota todo el árbol de directorios con sus diferentes ramificaciones, es la raíz, simbolizada por el carácter “/”. El investigador que accede por primera vez a Linux verá en este nivel una serie de directorios característicos. Cada uno de ellos posee una función y contiene archivos de un determinado tipo. Por ejemplo, en la raíz misma del árbol podrá verse el *kernel* del sistema, habitualmente con el nombre `vmlinuz` o algo parecido. En las distribuciones modernas el archivo no es más que un enlace al núcleo propiamente dicho que se encuentra en el directorio `/boot`, el cual contiene además otros programas necesarios para el arranque del sistema.

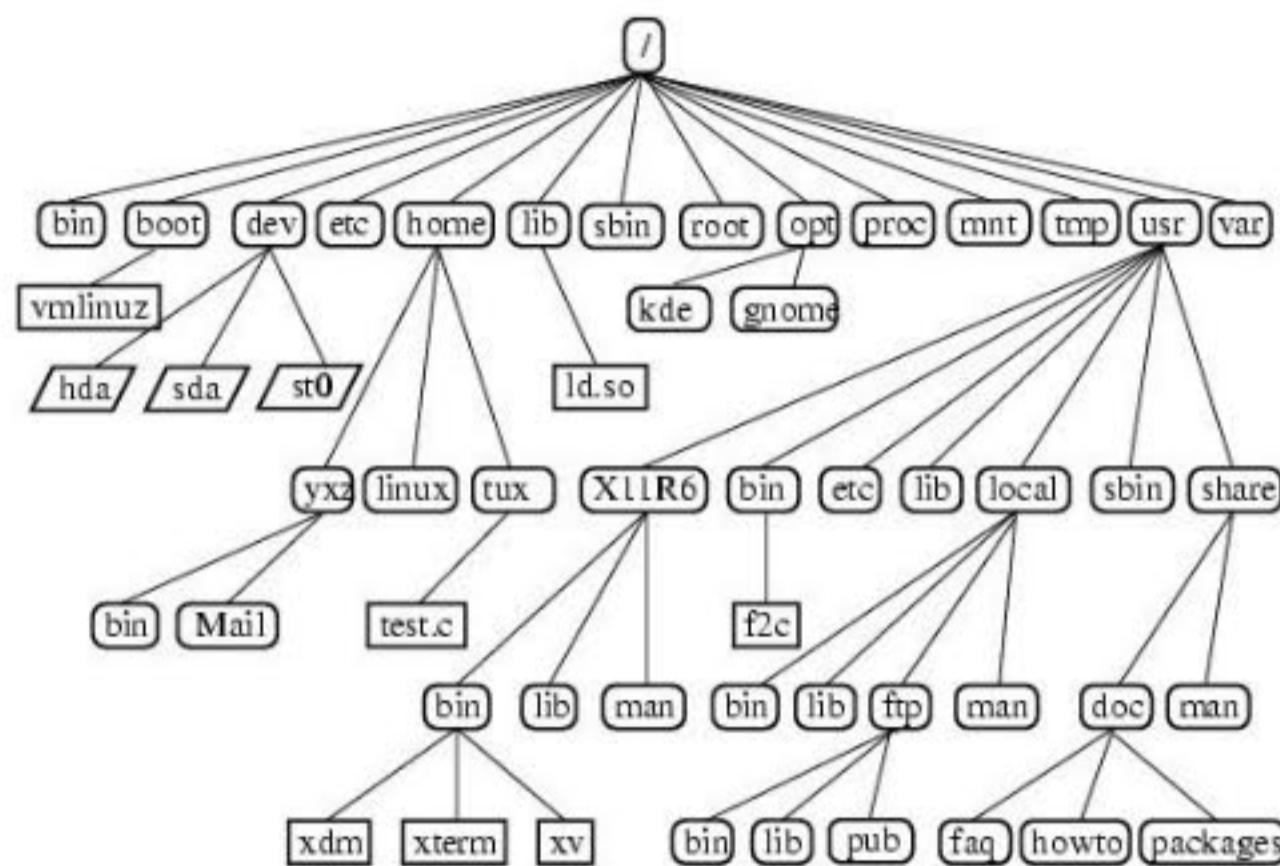


Figura 5.1. Árbol de directorios Linux (OpenSuSE Documentation, University of Cambridge)

En /bin se encuentran los ejecutables del sistema operativo. Se trata de los programas que el usuario utiliza para llevar a cabo todo tipo de operaciones desde la línea de comandos, como por ejemplo cd, ls, mv, mkdir, cat, dd, etc. Un directorio similar a este, /sbin, contiene binarios utilizados para fines de administración del sistema, como por ejemplo mount, shutdown, ps, etc. /dev contiene archivos que representan dispositivos del sistema: discos duros, terminales, unidades USB, etc. En /etc los diseñadores de Unix juzgaron conveniente alojar archivos de configuración para los programas instalados en el sistema. El directorio /lib contiene las bibliotecas y el código compartido que utilizan el sistema operativo y los programas. En /tmp se guardan los archivos temporales; el contenido de este directorio experimenta cambios cada vez que el sistema se reinicia. /mnt y /media sirven para el montaje de otros sistemas de archivos y dispositivos extraíbles como discos externos, llaves USB, discos ópticos, unidades ZIP, etc.

Finalmente hay cinco directorios que para el investigador poseen un interés particular: /proc, poblado por una estructura virtual de carpetas y archivos que documentan el estado de funcionamiento del sistema; /root, directorio base del superusuario o administrador del sistema; /usr, un árbol de directorios de gran importancia con software, bibliotecas y aplicaciones; /opt, utilizado por Linux para alojar paquetes adicionales de software; y finalmente /home, que contiene los subdirectorios, espacios de trabajo y documentos de los usuarios del sistema.

Debido a las restricciones de permisos necesarias para el funcionamiento y la seguridad de una estructura de directorios tan estandarizada y rígida es precisamente en /home donde el investigador probablemente hallará la mayor parte de los datos significativos para su análisis forense, como documentos, historiales bash, correo electrónico y bases de datos con la actividad de los navegadores de Internet. Esto sin embargo no quiere decir que el resto de las ubicaciones del sistema carezcan de interés. Más bien lo contrario, debido a la posibilidad de rootkits o manipulaciones en logs y archivos de configuración. Especial interés reviste el directorio del superusuario /root, en el que un intruso, de habérselas arreglado para conseguir privilegios de administrador, podría tener archivos guardados o haber instalado software malicioso. En última instancia, lo mismo que en Windows, es válido el principio de que la información puede hallarse en cualquier lugar del sistema.

Aunque los directorios se engarzan dentro de ubicaciones características dentro de la jerarquía estándar de Linux, eso no quiere decir que tengan que hallarse físicamente en la misma partición, ni siquiera en el mismo disco duro. De hecho y por conveniencias tanto de administración como de seguridad de funcionamiento, con frecuencia los directorios /home y /usr van instalados en particiones diferentes a la del resto del sistema, montándose en el árbol de directorios en el momento de iniciarse el sistema. Esto facilita las tareas de actualización del software de las aplicaciones al mismo tiempo que aporta una mayor seguridad al evitar que se mezclen en un mismo volumen el software del sistema y los datos de los usuarios.

### 5.2.3 Archivos y permisos

En un entorno Windows el usuario está acostumbrado a disponer de todos los objetos que componen el sistema como si fueran de su exclusiva propiedad y pudiera manipularlos a conveniencia. Aunque las versiones más avanzadas como Windows Vista y 7 restringen el acceso a los recursos del sistema –o por lo menos advierten al usuario antes de ejecutar una aplicación desconocida– conceptos como administrador, grupos o niveles de privilegio siguen siendo para la mayor parte de los usuarios poco más que simplemente teóricos y solo se aplican en redes de empresas y organizaciones públicas. Una vez instalado un sistema MS-Windows, sobre todo en entornos domésticos o en pequeñas empresas, al usuario le pasa lo mismo que a Mr. Jourdain, aquel personaje de Molière que se sorprendió al decirle que llevaba toda la vida hablando en prosa sin saberlo. Pues bien, la mayor parte de la gente maneja Windows XP o Vista con privilegios de superusuario sin tener la menor idea de lo que eso significa, con las inevitables repercusiones en términos de seguridad y estabilidad del sistema.

Linux es mucho más restrictivo en cuanto a la asignación de privilegios y la división de los usuarios en grupos. Esta división es más funcional que otra cosa. Lo que intenta no es separar a los usuarios en clases, sino hacer que cada uno de ellos asuma las competencias que en cada momento necesita para realizar sus tareas con el sistema sin comprometer la seguridad y el funcionamiento del mismo. Si hay un solo usuario y está trabajando en la redacción de un documento no hay ninguna razón para que lo haga con privilegios de administrador. Si por el contrario tiene que resolver algún problema técnico o instalar aplicaciones, asumirá dichos privilegios durante el tiempo preciso y después los abandonará volviendo a entrar al sistema como un usuario normal. Más que un intento de separar a las personas en clases, el esquema de privilegios de un sistema operativo moderno es como una de esas comedias en las que un mismo actor asume múltiples papeles, sin confundir jamás los guiones y haciendo medio mutis por el foro cada vez que deja de ser un personaje para transformarse en otro.

En general el usuario tendrá poder ilimitado sobre todos los archivos que haya en su zona de trabajo del directorio /home y sean de su propiedad, pero no se le permite trastear con los restantes, especialmente con aquellos que el sistema necesita para su funcionamiento. Habrá directorios a los que ni siquiera pueda acceder para listar su contenido. La única manera de hacerlo con el sistema en funcionamiento es ingresar como administrador tras haber abierto una consola *bash* o iniciado un navegador de archivos como Dolphin o Konqueror con privilegios de superusuario. Estas limitaciones no existen cuando el investigador lleva a cabo un análisis *post mortem*. Una vez montada la imagen del soporte podrá acceder sin problemas hasta el último recoveco del sistema de archivos.

Esta compartimentación la consigue Linux mediante un sistema de asignación de permisos. Cada archivo posee sus propios permisos, que figuran indicados en los *inodes* del sistema de archivos. Un archivo admite, por separado, permisos de lectura, escritura y ejecución para tres tipos de colectivos de usuarios, indicados por letras en las opciones del comando (*chmod*) que se utiliza para modificar privilegios y relaciones de propiedad: los propietarios del archivo (u), los miembros del grupo al que aquel pertenece (g) y todos los demás (o). En el listado de archivos de un directorio cada una de las entradas indica el propietario y los privilegios de acceso del archivo correspondiente: quiénes pueden leerlo, quiénes hacer cambios en él y quiénes ejecutarlo en el supuesto de que se trate de un programa o un *script*.

```
igandekoagigandekoa-desktop:~/Enlace$ ls -l
total 1177796
-rw-r--r-- 1 igandekoagigandekoa 1201596416 2011-12-26 14:04 eureka .avi
-rwxr--xr-x 1 igandekoagigandekoa 1322856 2010-10-04 18:04 busybox
-rw-r--r-- 1 igandekoagigandekoa 785123 2011-12-26 15:22 Contrato GOM.pdf
drwxr-xr-x 2 igandekoagigandekoa 12288 2011-12-28 15:57 Fotomontaje
drwxr-xr-x 2 igandekoagigandekoa 4096 2011-12-27 16:00 Joan_Sutherland
-rw-r--r-- 1 igandekoagigandekoa 760 2011-12-27 08:14 RW-DMVN-W5.txt
-rw-r--r-- 1 igandekoagigandekoa 1142238 2011-12-29 08:37 55DDFJ_V4_1_Lessard_Kessler.pdf
igandekoagigandekoa-desktop:~/Enlace$
```

Figura 5.2. Archivos y permisos

En la figura 5.2 se muestra un directorio con varios archivos. Dos de ellos son a su vez directorios (indicado por la letra “d” inicial). También vemos, además del nombre del propietario, dos documentos, un archivo de texto y otro Adobe PDF, que pueden ser manipulados de la manera siguiente: el propietario puede leerlos y modificarlos (primera secuencia de caracteres “rw-” inmediatamente después de la “d”); los miembros del grupo al que pertenece el propietario también pueden leerlos y modificarlos (segunda secuencia de caracteres “rw-”); cualquier otro solamente podrá leerlos (tercera secuencia de caracteres “r - -”). El archivo busybox es un ejecutable. La información relativa a sus permisos (-rwxr-x-r-x) indica que puede ser leído, modificado y ejecutado por su propietario; los miembros del grupo pueden leerlo y ejecutarlo pero no hacer cambios en él; cualquier otra persona podrá hacer lo mismo que el grupo. El administrador del sistema (*root*) está por su parte facultado para hacer lo que quiera con este archivo y con cualquier otro.

Existen dos tipos de permisos muy importantes para archivos ejecutables, Setuid y Setgid, que se indican mediante la letra “s” y sirven para que aquellos puedan ser utilizados con permisos de administrador por usuarios normales, en ocasiones excepcionales y para tareas muy específicas. Un archivo con el bit setuid activado figura de la manera siguiente en el listado del directorio:

```
-rwsr-xr-x 1 root shadow 27920 ago 15 22:45 /usr/bin/passwd
```

Aunque setuid es útil en algunos casos, también plantea riesgos de seguridad cuando este atributo se asigna a archivos binarios que no han sido bien programados o no pertenecen al sistema operativo. Un usuario con malas intenciones puede explotarlos para conseguir un nivel alto de privilegios o instalar troyanos en el sistema. Por los mismos motivos el investigador debe prestar una atención especial a la presencia de archivos setuid. El atributo setgid hace lo mismo que setuid pero en relación con los privilegios del grupo.

### 5.2.4 Marcas de tiempo

¿Por qué el denominado MAC-Time tiene una importancia tan grande en el análisis forense, sobre todo de sistemas Linux/Unix? MAC significa *modification, access* y *change*, respectivamente los momentos en que un usuario escribió un archivo por última vez (M), dicho archivo fue leído o ejecutado en caso de tratarse de un binario (A) o se realizó algún cambio en los metadatos de su *inode* (C). En Windows esta tercera marca de tiempo hace referencia al momento de la creación del archivo, y se actualiza al ser copiado este. En caso de mover el archivo o la carpeta donde aquel se encuentra la marca de tiempo no cambia.

Basándose en las marcas de tiempo el investigador puede averiguar de qué manera han sido manipulados los archivos en un ordenador sospechoso; a cuáles de ellos accedió el atacante, cuáles fueron alterados para propiciar una configuración del sistema favorable a las intenciones del intruso y si se instalaron troyanos o programas parásitos. Todo ello puede ayudar a esclarecer la forma en que se produjo la intrusión. La información MAC permite al investigador reconstruir, mediante gráficos o listados, una línea temporal que describa el desarrollo del incidente con sus momentos principales, añadiendo así un contexto preciso en el que situar todos los otros elementos de su análisis.

Las marcas de tiempo constituyen quizás el objeto más delicado de la evidencia, sobre todo cuando se trabaja con sistemas del tipo Linux/Unix. Esto se explica por un número de razones: en primer lugar el sospechoso puede alterarlas con facilidad o inutilizarlas mediante el empleo de técnicas antiforenses. Un investigador poco competente también puede modificarlas sin querer en el transcurso de su trabajo. La lectura del archivo o su ejecución –si se trata de un *script* o un comando del sistema– cambia la marca de acceso, perdiéndose de modo irreversible un dato cronológico anterior que puede resultar decisivo para el caso. Así mismo debe tenerse en cuenta que no todas las operaciones con archivos modifican de igual modo las marcas de tiempo. Mover un archivo o cambiarlo de nombre –acción que en Unix se lleva a cabo con el comando *mv* (*move*)– no afecta a ninguna de las tres marcas. El estudio de las marcas de tiempo en el análisis forense, por consiguiente, deberá hacerse desde una perspectiva crítica y sanamente escéptica y no sin antes haber aplicado precauciones para preservar en todo momento la integridad de la evidencia.

## 5.3 INFORMACIÓN VOLÁTIL

Teniendo en cuenta que el solo acto de listar un directorio puede alterar las marcas de tiempo de archivos que pueden constituir parte de nuestra evidencia, se hace ver la necesidad de proceder en el análisis forense de un sistema Linux de una forma metódica y más estricta aun si cabe que cuando se trabaja con Windows.

En el primer contacto con un sistema en funcionamiento y cuyo propietario haya tenido el acierto de dejar tal cual a disposición del investigador, sin reiniciarlo ni llevar a cabo indagaciones caprichosas o desordenadas en un intento de aclarar los hechos por su cuenta, hay que conceder prioridad a la búsqueda de informaciones valiosas para la investigación: marcas de tiempo, posible presencia de troyanos u otras piezas de software malicioso, usuarios dados de alta en el sistema, conexiones de red y procesos anómalos.

### 5.3.1 Fecha y hora del sistema

Fecha y hora del sistema son los primeros datos que se han de registrar en toda investigación forense que comience *in vivo*, con la máquina funcionando en el momento de llegar a ella la persona encargada de llevar a cabo la primera intervención. No se trata de un mero trámite para añadir al informe. Ese momento exacto, que naturalmente habrá que corregir en caso de discrepancia con una fuente fiable –otra de las obligaciones del investigador forense: llevar su reloj siempre en hora al igual que los ferroviarios–, marca un preciso límite para diferenciar entre las manipulaciones realizadas por el sospechoso y las que puedan derivarse de las posteriores operaciones de análisis llevadas a cabo por el investigador.

Tanto la fecha como la hora del sistema se obtienen mediante el comando *date*. He aquí un ejemplo:

```
igandekoa@igandekoa-desktop:~$ date
vie dic 30 18:30:48 CET 2011
```

### 5.3.2 Información de interés

No existe una regla invariable a la hora de rescatar información volátil de un sistema en funcionamiento. El investigador debe tener en cuenta que su trabajo es más eficaz si extrae los datos en un orden de prioridad inverso al de su persistencia en el sistema. Primero debe asegurar lo que más probabilidades tenga de verse alterado con el paso del tiempo, con un orden de prioridad que dependiendo de las circunstancias podría ser más o menos este: la caché, el contenido de la memoria RAM, las conexiones de red (puertos abiertos, aplicaciones que las utilizan, direcciones IP remotas) y los procesos en ejecución (procesos padre, propietarios, tiempos de ejecución, parámetros de comando, etc.). No olvide que el directorio */proc* contiene los binarios de los programas en curso, aunque hayan sido eliminados o interrumpidos, junto con información relativa a los mismos. He aquí un ejemplo de información obtenida sobre un proceso cualquiera a partir de los datos existentes en */proc*:

```
# cat /proc/PID/exe > results.txt
```

```
# strings results.txt
```

Sustituya PID por el número de proceso correspondiente. Para obtener el resto de la información volátil existen un número de comandos útiles que se presentan resumidos en la tabla 5.1. Estos comandos probablemente están ya presentes en el sistema investigado en forma de programas ejecutables dentro de los directorios /bin y /sbin, pero no debería recurrirse a ellos debido a la probabilidad de que estén troyanizados y puedan proporcionar informaciones falseadas. El investigador deberá llevar sus propias versiones compiladas estáticamente y ejecutarlas desde un medio externo de solo lectura, como por ejemplo un CD-ROM o una llave USB con el protector de escritura activado:

hostname	Nombre del sistema en red
cat /proc/cpuinfo	Información relativa al microprocesador
cat /proc/meminfo	Memoria RAM del sistema
uname -a	Versión y fecha de liberación del <i>kernel</i> de Linux
cat /etc/fstab	Puntos de montaje en el árbol de directorios
cat /proc/swap	Información sobre particiones y archivos de intercambio
cat /proc/modules	Módulos activos utilizados por el <i>kernel</i>
df -h	Información sobre particiones, tamaño y grado de ocupación de las mismas
env	Variables de entorno
who	Usuario activo
ifconfig -a	Interfaces de red
netstat -anp	Conexiones de red activas
netstat -rn	Tabla de enrutamiento del <i>kernel</i>
lsof	Archivos abiertos por procesos en ejecución
nsone -P -i -n	Puertos abiertos
ps -e	Lista de procesos en ejecución y un número de parámetros relativos a los mismos
cat /etc/hosts	Muestra el contenido del archivo hosts con la equivalencia prioritaria entre nombres de dominio y direcciones IP
cat /etc/passwd	Archivo de contraseñas
cat /etc/shadow	Archivo shadow con los <i>hashes</i> de las contraseñas
arp -n	Entradas de la tabla Arp
cat /etc/resolv.conf	Configuración de los DNS

Tabla 5.1. Comandos para extraer información volátil en Linux

### 5.3.3 Puertos y conexiones abiertas

Un caso típico de manipulación se produce cuando el intruso ha instalado un troyano para poder acceder al ordenador desde una ubicación remota. El procedimiento para descubrirlo es el mismo que en Windows: examinar las conexiones abiertas para ver si hay alguna dirección IP que nos parezca anómala conectada a un puerto que tampoco sea de los que el sistema utiliza para su funcionamiento habitual ni esté documentado en los procedimientos de la empresa. A continuación comprobamos qué procesos se conectan a ese puerto, cuál es la localización de los archivos correspondientes y si puede haber alguien que intente disimular su presencia en el sistema, por ejemplo nombrando a sus ejecutables como si fueran archivos del sistema o utilizando caracteres especiales.

Este método, que el autor ha empleado en sistemas Windows para localizar troyanos y otros programas parásitos particularmente resistentes a los antivirus, también da buenos resultados en Linux. Existe la posibilidad de que el sistema esté infectado por un *rootkit* capaz de falsear la información proporcionada por los comandos. Por ese motivo se recomienda utilizar programas compilados estáticamente desde un CD en lugar de las herramientas locales del sistema. Aun así no existen garantías absolutas de obtener una información fiable al 100%, ya que podría tratarse de un *rootkit* de núcleo con capacidad para alterar el flujo de datos e instrucciones a un nivel de ejecución más bajo que aquel en el que funcionan las librerías y las aplicaciones.

He aquí un ejemplo de rastreo de conexiones abiertas:

```
# sudo /media/CDForensics/tools/netstat -plutn >
/media/USBForensics/caso1/conexiones.txt
```

En la última columna de la tabla resultante se puede apreciar el nombre del programa y su número de proceso. En este ejemplo concreto (figura 5.3) observamos, entre otras aplicaciones, un cliente bittorrent en funcionamiento con dos puertos abiertos: 6881 (para las conexiones TCP) y 4444 (UDP). Si no es de los que apuntan cosas a mano en una libreta acostúmbruese a redirigir la salida de los comandos a archivos de texto ubicados en un soporte de datos externo o a otro *host* de la red mediante el comando netstat -j ¡jamás utilice el disco duro del ordenador sometido a investigación!-. Si tiene conocimientos de *bash*, para facilitar su tarea, también puede programar un *script* con todas las órdenes que figuran en la tabla 5.1.

```
igandekoa@igandekoa-desktop:~$ sudo netstat -pltn
Conexiones activas de Internet (solo servidores)
Proto Recib Enviad Dirección local     Dirección remota   Estado      PID/Program name
tcp    0      0 127.0.0.1:631          0.0.0.0:*        ESCUCHAR    987/cupsd
tcp    0      0 127.0.0.1:25         0.0.0.0:*        ESCUCHAR    1358/exim4
tcp6   0      0 ::1:6881            ::*:             ESCUCHAR    2092/ktorrent
tcp6   0      0 ::1:631             ::*:             ESCUCHAR    987/cupsd
tcp6   0      0 ::1:25              ::*:             ESCUCHAR    1358/exim4
udp    0      0 0.0.0.0:5353        0.0.0.0:        922/avahi-daemon: r
udp    0      0 0.0.0.0:33683       0.0.0.0:        922/avahi-daemon: r
udp    0      0 0.0.0.0:68          0.0.0.0:        946/dhclient
udp6   0      0 ::::44776           ::*:             922/avahi-daemon: r
udp6   0      0 ::::5353            ::*:             922/avahi-daemon: r
udp6   0      0 ::::4444            ::*:             2092/ktorrent
igandekoa@igandekoa-desktop:~$
```

Figura 5.3. Conexiones abiertas

### 5.3.4 Procesos en ejecución

Obtener la tabla de procesos mediante ps es trivial. Sin embargo el investigador deberá comprobar el resultado examinando con todo cuidado el archivo de texto resultante. Los procesos parásitos tienen la mala costumbre de camuflarse bajo los nombres de otros legítimos. A veces se sirven también de caracteres extraños para confundir al investigador. Si encuentra algo que parece estar fuera de su sitio vale la pena mirar con atención. Especialmente sospechoso es todo aquello que parece estar ejecutándose desde ubicaciones no conformes con el estándar FHS o tenga relación con netstat. El investigador deberá combinar los comandos de administración del sistema, utilizando sus opciones con el objeto de obtener informaciones cruzadas que le permitan poner al descubierto posibles anomalías.

## 5.4 ADQUISICIÓN FORENSE

Una vez rescatada la información volátil es hora de apagar el sistema, pero no de una manera ordenada con el comando shutdown sino por el método tradicional en las investigaciones forenses, cortando la corriente para evitar cambios adicionales por efecto de las rutinas de apagado u otras que pudieran estar programadas con el objeto de eliminar información comprometedora entre una sesión y la siguiente. En principio la adquisición de los sistemas Linux/Unix no difiere de la de uno de tipo MS-Windows. Se emplea incluso el mismo software –normalmente EnCase–. Toda plataforma informática basada en Unix dispone no obstante de herramientas gratuitas o de bajo coste y fácil instalación que permiten realizar copias a bajo nivel de discos duros y otros soportes de datos.

### 5.4.1 Adquisición con dd

Cuando se trata de un disco con particiones Unix y este ha de ser adquirido desde un sistema Linux –o Apple OSX–, la primera opción que se presenta es el comando dd o uno de los derivados del mismo que vimos en el capítulo 3. Se pueden adquirir discos enteros o particiones separadas. En este último caso el investigador deberá tener cuidado para no dejar fuera del proceso de adquisición las zonas del soporte no particionadas.

```
dd if=/dev/sda of=/imagen.dd conv=noerror, sync bs=4096k
```

Esto permite adquirir un disco entero. Si el investigador quiere las particiones separadas deberá extraer una imagen de bajo nivel para cada una de ellas, introduciendo órdenes sucesivas como por ejemplo:

```
dd if=/dev/sda1 of=/particion1.dd conv=noerror, sync bs=4096k,
```

```
dd if=/dev/sda2 of=/particion2.dd conv=noerror, sync bs=4096k,
```

...etcétera.

La opción conv=noerror especifica que dd debe continuar con la copia aunque encuentre errores de lectura que en casos normales le obligarían a detener el proceso; sync rellena las partes ilegibles con ceros al objeto de conseguir una imagen del mismo tamaño que el original. No olvide que la forma de nombrar unidades y particiones es distinta dependiendo de si se trata de discos PATA (hda0, hda1, hdb0, hd1) o SATA (sda0, sda1, etc.) y también de unas distribuciones Linux a otras (en Ubuntu una unidad PATA aparece denominada siempre como sdaX). Consulte la documentación de su sistema antes de hacer nada. Para realizar adquisiciones forenses es probable que el sistema le pida asumir privilegios de superusuario. No olvide que a la hora de utilizar dd –sobre todo si pretende llevar a cabo la copia sin un bloqueador de escritura– debe fijarse bien en lo que teclea. Confundir accidentalmente la fuente y el destino en una adquisición arruinaría no solamente la evidencia sino también el sistema investigado, provocando una catástrofe jurídica irreparable.

En numerosos artículos sobre técnicas de adquisición forense el lector podrá encontrar consejos relativos a las opciones de dd, incluida la conveniencia de establecer mediante “bs” un tamaño de bloque que se corresponda con el del sistema de archivos para hacer más rápido el proceso de copia a bajo nivel. Este requisito podía ser importante con los lentes y rudimentarios discos duros de hace

diez años, pero en la práctica carece de relevancia para soportes modernos. El autor ha realizado pruebas con discos duros posteriores al año 2000 utilizando tamaños de bloque diferentes y no ha notado disparidades apreciables en el tiempo de adquisición.

#### 5.4.2 Adepto

Adepto de *e-fense.com* forma parte de la *suite* forense Helix. Esta herramienta fue diseñada para la adquisición de imágenes en flujo de bits y la gestión de cadenas de custodia. Básicamente consiste en un *front end* o interfaz gráfico para las ya explicadas utilidades Unix en línea de comando dd/dcfldd/sdd, con un número de opciones adicionales que lo convierten en el reemplazo ideal de otras utilidades ya obsoletas como AIR o GRAB. Adepto dispone entre otras de las funcionalidades siguientes:

- Autodetección de discos IDE/SATA y SCSI, CD-ROM y unidades de cinta.
- Posibilidad de elegir entre dd, dcfldd o sdd.
- Comprobación de imágenes mediante *hash* MD5/SHA1.
- Realización de imágenes a través de Netcat/Cryptcat o NetBIOS (SAMBA).
- Compresión y descompresión de datos por medio de utilidades Unix estándar como gzip/bzip2.
- Limpieza de soportes y particiones (cubriendo con ceros).
- Registro detallado de horas, fechas y actividad en línea de comando.
- División de imágenes en archivos más pequeños y manejables.

El proceso es gradual y se estructura en una secuencia de pasos estrictos, lo cual ayuda al investigador a ejecutar la tarea de adquisición con eficacia y de manera metódica. La pantalla inicial ofrece al investigador un menú para que este introduzca su nombre de usuario y un número de caso para finalidades de administración. Tras haber cumplido este trámite el programa permite el acceso a varias pestañas desde las cuales se pueden ajustar todos los parámetros del software

para la adquisición. La pestaña de información de dispositivos (Device Info) permite examinar y seleccionar los diferentes soportes conectados al sistema, con la posibilidad de consultar datos de marca, modelo, características, etc., para incluirlos en el informe. Si algún medio no figura en la lista se puede activar un reescaneo de dispositivos.

Una vez seleccionado el dispositivo se permite el acceso a la pestaña de adquisición, desde la cual el investigador decide cómo llevar a cabo la copia, incluyendo anotaciones relativas al caso y otros datos de interés. En este apartado es donde se especifica si la adquisición ha de realizarse mediante dd o alguno de sus clones, qué tipo de *hash* aplicar, el tamaño de los archivos de destino, o bien si la copia es local o ha de ser transferida a otro ordenador de la red mediante Netcat o Samba. Una vez seleccionadas las opciones, el usuario debe activar **Start** para iniciar el proceso de adquisición. En caso de utilizar Netcat/Cryptcat también deberá introducir la dirección IP de destino y el puerto en el que permanece a la escucha el Netcat/Cryptcat que está funcionando como servidor. Samba y NetBIOS requieren previamente un montaje del volumen remoto. Finalmente las pestañas de informes (Log) y cadena de custodia (Chain of Custody) sirven para generar documentos automatizados sobre el proceso de copia, que se guardan junto con la imagen a bajo nivel en la unidad de destino. El investigador no tiene más que introducir el número asignado al objeto de evidencia y hacer clic sobre el botón **Create**.



Figura 5.4. Adepto

Puesto que Adepto es un *front end*, es decir, un interfaz gráfico a través del cual se manejan las utilidades en línea de comando que hacen el trabajo duro, en teoría el usuario ni siquiera tiene que conocer la sintaxis de dd y sus otras variantes para la consola *bash*. Ha de insistirse en que es importante no confundir las unidades de origen y de destino. Adepto está incluida en algunas distribuciones Linux de seguridad y forenses y es la herramienta de adquisición estándar del CD autoarrancable Helix 3.

## 5.5 ANÁLISIS

Al igual que sucede con cualquier otro sistema, en Linux el análisis de la evidencia depende de la naturaleza del caso, las tesis del Ministerio Fiscal y otras circunstancias. Si se tienen indicios de la presencia de material ilícito habrá que buscar archivos ocultos o borrados. Si creemos que el ordenador ha sido utilizado para cometer algún tipo de actividad ilegal llevaremos a cabo búsquedas de caracteres, registros de correo electrónico o un análisis del historial de Internet. Y, finalmente, si sospechamos que la máquina ha podido ser comprometida y en ella hay instalada una puerta trasera o un *rootkit*, habrá que localizar rastros de su actividad examinando los *logs* y verificando la presencia de archivos extraños en el sistema.

### 5.5.1 La línea de tiempo

La línea de tiempo constituye un elemento de la mayor importancia a la hora de investigar cualquier actividad irregular en un sistema. Se trata de una de esas cosas que tendemos a considerar accesorias o meramente ilustrativas pero en la práctica resultan imprescindibles para evitar ataques ante el tribunal. Sin un esquema preciso de orden temporal todas las afirmaciones del investigador son relativas y vulnerables a los argumentos de una parte contraria que posea habilidad suficiente para explotar las discrepancias a las que da lugar una presentación desordenada de los hechos.

No existen reglas precisas para elaborar una línea de tiempo, aparte de la coherencia y una estricta regularidad cronológica en la presentación de los hechos detectados. El procedimiento habitual consiste en utilizar una hoja de papel del mayor tamaño posible, colocarla en posición apaisada y señalar los acontecimientos del caso (encendido del sistema, mensajes de error, creación de determinados archivos, momento en que el administrador avisó a las autoridades del incidente, etc.) sobre una línea horizontal de modo que resulten legibles y transmitan una imagen claramente consecutiva de los acontecimientos.

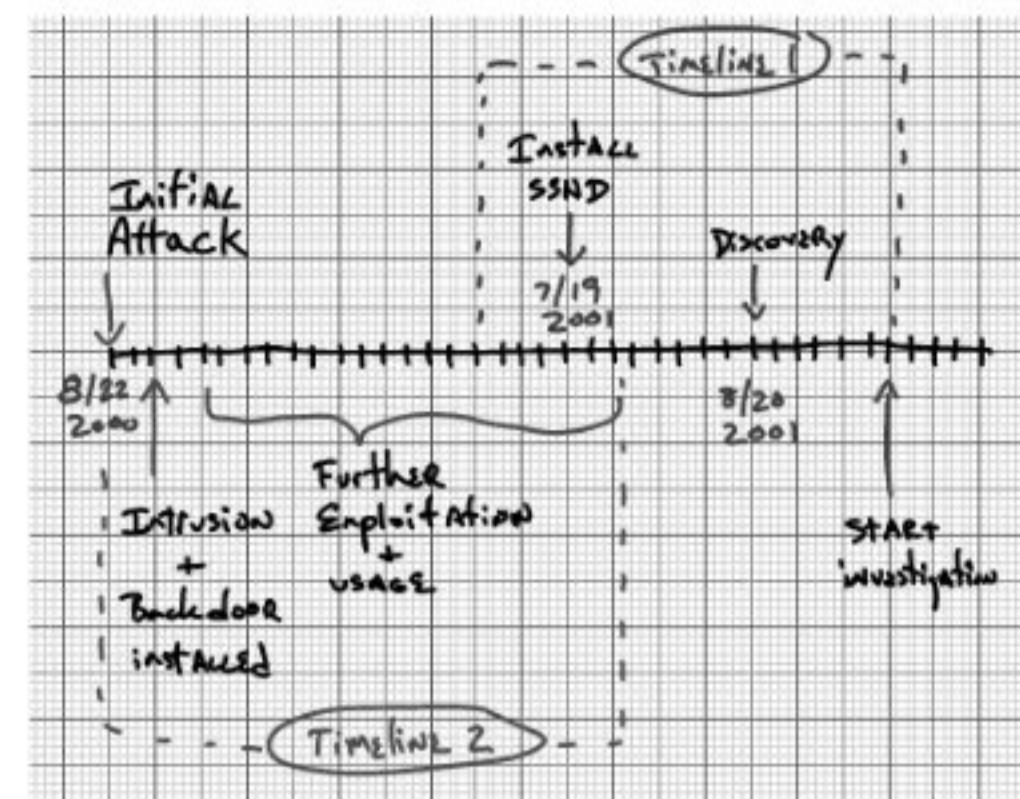


Figura 5.5. Ejemplo de línea de tiempo (Farmer, Venema: *Forensic Discovery*. Addison-Wesley, 2005)

Tampoco existe un software capaz de elaborar una línea de tiempo a partir de la imagen en flujo de bits adquirida por el investigador. La diversidad de criterios y la presencia de hechos que tienen lugar más allá del teclado del ordenador dificultan la tarea de programación, haciéndola prácticamente imposible. Pasar de la adquisición de evidencias a la realización de un esquema temporal supone un salto cualitativo solo comparable a la diferencia entre un funcionario que aplica una serie de procedimientos establecidos y el científico que intenta buscar una teoría plausible a partir de una constelación de datos desordenados. La única ayuda mecanizada de la que se dispone son herramientas capaces de poner los archivos en orden cronológico después de haber examinado sus marcas de tiempo. Y esto, naturalmente, con las oportunas reservas, a las que ya se ha hecho mención, sobre la condición frágil y manipulable de los datos digitales.

### 5.5.2 Herramientas para elaborar una línea de tiempo

Vamos a ver cómo se puede obtener, con herramientas nativas y de código libre, un “borrador” para nuestra línea de tiempo en entornos Linux/Unix. Para ello nos serviremos de la utilidad fls perteneciente a la *suite* forense TSK, que tuvimos ocasión de conocer en el capítulo 3. Este comando nos permite obtener a partir de la imagen a bajo nivel un listado de archivos con sus marcas de tiempo. Por ejemplo:

```
# fls -o 63 -f linux-ext2 -m / -r imagen.dd > listado.flx
```

En este caso nuestra evidencia consiste en la imagen completa de un disco particionado según el sistema habitual MSDOS –incluyendo MBR y cuatro particiones primarias–. Por eso especificamos el salto de 63 sectores hasta el comienzo de la partición. De otro modo fls no sería capaz de detectar el sistema de archivos. Utilizamos la opción `-f linux-ext2` para especificar que se trata de un sistema de archivos Linux típico, información que ya conocemos por haber llevado a cabo previamente un análisis de la imagen con mmls o fdisk. La opción “m” avisa a la herramienta para que incluya en el listado las marcas de tiempo de los *inodes* de los archivos, comenzando a partir del directorio `/`.

A continuación sometemos nuestro archivo de texto con los resultados de fls a la acción de mactime, otra herramienta TSK:

```
# mactime -b listado.fls > timeline.txt
```

mactime clasifica los resultados obtenidos con fls y los presenta en una forma legible. El *output* de mactime suele ser muy voluminoso. De ahí la necesidad de redirigirlo a un archivo de texto que después podremos examinar con un editor o un programa de hojas de cálculo.

Numerosos sistemas actualizan las marcas de tiempo en el borrado de archivos. Por este motivo, si el investigador sospecha que parte de la evidencia puede haber sido borrada, debería hacer también otro listado con ils:

```
# ils -o 63 -f linux-ext2 -m > listado2.fls
```

A diferencia de fls, ils muestra directamente la información de los metadatos, haciendo posible reconocer procesos de borrado que de otro modo pasarían desapercibidos si utilizamos únicamente fls.

### 5.5.3 Recuperación de archivos borrados

Antes de recuperar masivamente los archivos borrados del sistema, con la ayuda de una *suite* comercial como EnCase o FTK o mediante herramientas de *data carving* como Foremost o Testdisk/Photorec, el investigador debe averiguar qué archivos han sido borrados, principalmente si el caso tiene que ver con accesos no autorizados o instalación de troyanos. Para obtener una lista de archivos borrados en un directorio puede recurrir a fls. La opción `-d` muestra únicamente los archivos borrados. Con `-r` el proceso se hace recursivo para todos los subdirectorios:

```
# fls -o 63 -f linux-ext2 -r -d imagen.dd > deleted.txt
```

El primer carácter indica qué tipo de objeto es el archivo borrado: directorios (d), archivos (r) o enlaces simbólicos a otros archivos o directorios (l). El tercer carácter –después del asterisco– muestra el *inode* correspondiente. A continuación vienen el nombre y la ruta completa del archivo. Podemos intentar recuperarlo con TSK a condición de que el sistema no haya experimentado un nivel de actividad demasiado alto con posterioridad al borrado. Para ello en primer lugar averiguamos con istat el tamaño del archivo sospechoso y los bloques de datos pertenecientes al mismo:

```
# istat -o 63 imagen.dd [Número de Inode]
```

Acto seguido lo recuperamos con la ayuda de icat:

```
# icat -o 63 imagen.dd [Número de Inode] > archivo.icat
```

Esta operación extrae los bloques que el sistema de archivos todavía tiene asignados al archivo borrado y los agrupa en otro archivo que podrá llevar el nombre que queramos ponerle. Si los bloques de datos están intactos nada impide que el archivo recuperado sea utilizado con el mismo grado de funcionalidad y las mismas características que tenía antes de ser eliminado: documento, imagen, ejecutable binario. Si sospechamos que este último puede ser el caso, no está de más tomar precauciones. No intente ejecutar ningún archivo sospechoso recuperado de imágenes forenses, sobre todo si está trabajando con privilegios de administrador. Analícelo a fondo para estar seguro de que no contiene código capaz de comprometer su evidencia o desestabilizar su sistema operativo. Un buen punto de partida consiste en tratar de identificar el archivo mediante el comando file y la información contenida en el archivo de firmas Magic. A continuación debe examinar el contenido del archivo sospechoso con el comando strings. Esta operación puede llevarse a cabo tanto sobre el archivo recuperado como directamente sobre la imagen forense:

```
# icat -o 63 imagen.dd [Número de Inode] | strings
```

## 5.6 OTRAS HERRAMIENTAS

### 5.6.1 Chkrootkit y Rkhunter

La peor forma de comprometer un sistema informático es mediante la instalación de un *rootkit*. Probablemente el lector ha oído hablar de estos sofisticados y malévolos artefactos que constituyen un peligroso refinamiento en el desarrollo de software maligno. A diferencia de los virus y troyanos de corte tradicional, el objetivo de un *rootkit* no consiste en infligir daños masivos a los usuarios, sino en disimular la existencia de vías de acceso con un nivel alto de

privilegios a un ordenador. Para ello proporciona información falseada sobre el funcionamiento normal de las aplicaciones y el sistema operativo. Por lo general el *rootkit* es instalado por el intruso aprovechando la primera ocasión que se le presenta. Los objetivos pueden ser varios: convertir el *host* en plataforma de ataque contra otras redes, robo de información o quizás tan solo espia a los empleados de la empresa. El usuario legítimo del ordenador continúa trabajando con normalidad sin advertir el menor indicio de que su sistema ha sido comprometido.

```

Checking 'sniffer'...
r sockets
eth0: PACKET SNIFFER(/sbin/dhcclient3[978])
Checking 'w55808'...
Checking 'wted'...
Checking 'scalper'...
Checking 'slapper'...
Checking 'z2'...
ed from lastlog!
Checking 'chkutmp'...
ss(es) were not found
in /var/run/utmp !
! RUID      PID TTY      CMD
! root      1013 tty7  /usr/bin/X :0 vt7 -nr -nolisten tcp -auth /var/run/xauth/A:0-GqUw7a
chkutmp: nothing deleted
Checking 'OSX_RSPLUG'...
not infected
igandekoa@igandekoa-desktop:~$ 

```

Figura 5.6. Chkrootkit

Listar archivos o procesos resulta inútil porque el *rootkit* oculta todo aquello que podría parecer sospechoso. Esto lo consigue reemplazando ejecutables y librerías que funcionan en el espacio de usuario por versiones modificadas capaces de ocultar la presencia del software maligno que controla el sistema. Algunos *rootkits* llegan más abajo, hasta el anillo principal, alterando el flujo de ejecución del *kernel*, con lo cual un sistema puede quedar comprometido aunque todas sus aplicaciones y librerías originarias se encuentren intactas. Finalmente en los últimos años se han desarrollado *rootkits* que son capaces de comprometer el hardware que gestiona los hipervisores de las máquinas virtuales en algunas CPU de reciente desarrollo, con lo cual resultan prácticamente indetectables.

Existen algunas herramientas útiles para la detección de *rootkits* en entornos Linux/Unix. Chkrootkit y Rkhunter son *scripts* que utilizan comandos usuales de Linux como strings o grep para localizar cadenas de texto sospechosas en los programas utilizados por el administrador del sistema operativo. También llevan a cabo una comparación con referencias cruzadas entre la salida del comando ps y la información existente en el directorio virtual de procesos /proc en busca de discrepancias que pudieran ser reveladoras de un intento de falseamiento. Por su parte Rkhunter compara los *hashes* SHA1 de los programas principales del sistema con una base de datos segura para comprobar si alguna utilidad ha podido ser alterada o reemplazada por un parásito.

```

Checking for passwordless accounts [ None found ]
Checking for passwd file changes [ None found ]
Checking for group file changes [ None found ]
Checking root account shell history files [ None found ]

Performing system configuration file checks
  Checking for SSH configuration file [ Not found ]
  Checking for running syslog daemon [ Found ]
  Checking for syslog configuration file [ Found ]
  Checking if syslog remote logging is allowed [ Not allowed ]

Performing filesystem checks
  Checking /dev for suspicious file types [ Warning ]
  Checking for hidden files and directories [ Warning ]

[Press <ENTER> to continue]

```

Figura 5.7. RKhunter

Estas herramientas se ejecutan con privilegios de administrador. Su funcionamiento es simple. En el caso de Chkrootkit basta con teclear el nombre del programa en una consola de texto:

```
# sudo chkrootkit > informe.txt
```

Rkhunter genera por defecto un archivo de informe en el directorio /var/log. Para crear el mencionado archivo en una ubicación es preciso especificarlo mediante la opción “-l”. Por ejemplo:

```
# sudo rkhunter --check -l /media/USB/resultados.txt
```

## 5.6.2 Md5deep

La suma de verificación no solo permite asegurar la cadena de custodia de la evidencia, sino también localizar cambios no autorizados en un archivo. En determinadas circunstancias conviene disponer de una herramienta capaz de obtener *hashes* de manera recursiva, que se pueda emplear de manera versátil y ahore al investigador el trabajo de crear sus propios *scripts*. Md5deep se encarga de esto: analiza todos los archivos de un árbol de directorios elaborando una lista de los *hashes* correspondientes que después podrá ser utilizada para verificar cualquier intento de manipulación. El modo de empleo es el siguiente:

```
# md5deep -rl "directorio" > resultados.md5
```

Con lo anterior se obtiene una lista de *hashes* “autorizada”. Las opciones utilizadas significan que la herramienta tiene que trabajar recursivamente (r) y registrar la ruta completa del archivo para facilitar su identificación. Posteriormente, y para verificar la integridad de los archivos, puede hacerse esto:

```
# md5deep -X resultados.md5 -r directorio/
```

Al haber sido desarrollado en el ejercicio de sus funciones por Jeff Kornblum, investigador forense que trabaja para el servicio de seguridad informática de la Marina de Estados Unidos, md5deep es propiedad pública. Cualquier usuario puede utilizarlo sin necesidad de solicitar licencias ni autorizaciones de ningún tipo.

## REDES E INTERNET

Investigar en entornos de red no es tarea fácil. Las redes informáticas constituyen un campo de gran complejidad que además de lo enrevesado de la tecnología subyacente añade a la actividad del investigador otras dificultades: presencia de nodos que dificultan la búsqueda de evidencia y de máquinas sospechosas, un caudal de datos ingente ininterrumpido, carácter volátil de la información e imposibilidad de aplicar metodologías estándar basadas en la adquisición de soportes y el análisis *post mortem*. Hasta ahora el investigador podía sentirse relativamente cómodo porque se enfrentaba a tareas abarcables que podían resolverse mediante herramientas de software cuya potencia procede en gran parte del hecho de haber dispuesto de tiempo para afinarlas durante unos años en los que la tecnología básica –discos duros, sistemas de archivos, interfaces de conexión, sistemas operativos– apenas experimentó cambios significativos. Sabía dónde estaba el objeto de sus pesquisas y lo que había que hacer con él, y disponía de métodos precisos para asegurar la cadena de custodia y llevar a cabo una labor de investigación con garantías suficientes. Ahora está a punto de entrar en un dominio de alto riesgo en el que las innovaciones se suceden con rapidez, se amontonan las dificultades y los resultados del trabajo no siempre son previsibles o de clara interpretación.

### 6.1 COMPONENTES DE UNA RED

Antes de conocer las tecnologías y herramientas que ha de utilizar, el investigador deberá familiarizarse con lo básico del análisis forense de redes. Para ello se ha resuelto dividir este capítulo en tres apartados principales: en primer lugar los componentes principales de una red corporativa; acto seguido la

### Capítulo 6

importancia de los archivos de registro (*logs*) en los que se guardan los datos que interesan para su investigación, y finalmente la vigilancia e intercepción del tráfico y la preservación de elementos de evidencia.

#### 6.1.1 Visión general de una red corporativa

Rara vez un servidor es atacado directamente. Para llegar a él con los fines que sea (robar información, instalar software maligno, comprometerlo en una agresión contra otras redes, etc.), antes el intruso, a no ser que disponga de acceso directo al cuarto de los servidores o un buen juego de ganzúas, habrá tenido que sortear un número de componentes funcionales implementados en hardware y software. Si se trata de un empleado de la empresa quizás se haya conectado externamente a través de una red virtual. Y si es un delincuente informático habrá tenido que desplegar técnicas de penetración o engañar a alguien del personal para que instale en su ordenador un troyano. En cualquier caso habrá tenido que atravesar uno o varios *routers* y entendérselas con un *firewall*, un sistema de detección de intrusos o cualesquiera otras características de seguridad implementadas en los modernos dispositivos de conexión de redes. Si estos nodos tienen habilitados sus sistemas de registro, en los *logs* correspondientes pueden haber quedado rastros de la actividad del atacante.

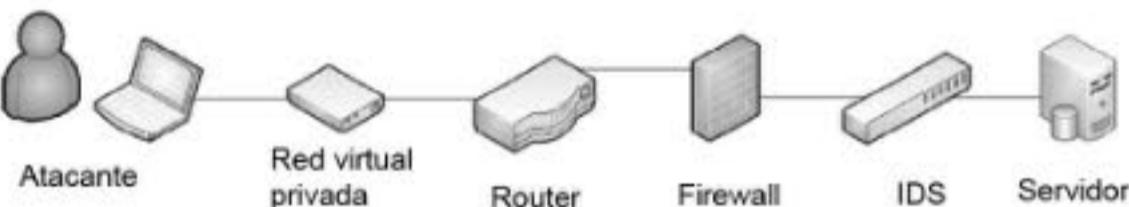


Figura 6.1. Esquema de una red local vista por el agresor

#### 6.1.2 Archivos de registro

En una red corporativa existen aplicaciones con capacidad para registrar lo que hace el usuario: sistemas operativos, servidores DHCP, servidores de páginas web, cortafuegos, servidores de autenticación, IDS e incluso *logs* de Netflow, un sistema de auditoría presente en la mayor parte de los enruteadores modernos. Esto no quiere decir que la actividad del usuario quede registrada sin fisuras. Algunas dificultades entorpecen la búsqueda de evidencia. En primer lugar, el carácter volátil de la información. Las redes informáticas son dinámicas por naturaleza. El tamaño de los *logs* es grande pero no ilimitado. El elevado flujo de datos que soporta una red hace que a no ser que la intervención del investigador sea inmediata, la evidencia haya podido quedar eliminada irreversiblemente tras el inicio de un nuevo ciclo de rotación de los *logs*. Además hay que contar con la

posibilidad de que el intruso haya modificado o borrado archivos de registro para ocultar su acción, o que el sistema de *logs* esté deshabilitado para ahorrar espacio en los discos duros o las memorias de estado sólido de los *routers*.

Sin embargo, con todas sus limitaciones, las entradas en los archivos de registro constituyen el principal elemento de evidencia accesible al investigador. Por ejemplo, esta es la entrada *lease* de un servidor DHCP.

```
Lease 192.168.21.13 {
    starts 0 2011/10/02 19:31:02;
    ends 1 2011/10/02 19:53:12;
    hardware ethernet 00:e0:98:65:2a:6b;
    uid 01:00:e0:98:65:2a:6b;
    client-hostname "philo";
}
```

DHCP es un servicio que permite la conexión a un punto de acceso inalámbrico o un *router* sin tener que configurar direcciones IP ni un servidor de nombres. Basta introducir el cable en el conector de la tarjeta de red o poner en marcha un dispositivo con interfaz WiFi en las proximidades de un punto de acceso inalámbrico para que DHCP, después de un breve intercambio de señales, asigne una dirección IP al *host*, con la que este podrá identificarse dentro de la red. Si el sistema de *logs* del servidor DHCP está habilitado quedará un rastro de cada una de las conexiones.

En este caso el *lease* del DHCP aporta informaciones que dependiendo del contexto de investigación pueden tener interés forense: un ordenador con nombre “philo” se unió a la red el 2 de octubre de 2011. Tras haberle sido asignada la dirección IP de red local 192.168.21.13, permaneció conectado aproximadamente durante veinte minutos. La dirección MAC del interfaz Ethernet –un número de 48 bits asignado en virtud de convenios internacionales y único en todo el mundo para cada dispositivo– permite establecer la identidad del ordenador, y posiblemente también la del usuario.

### 6.1.3 Preservación de elementos de evidencia en redes

Para entender las dificultades de la investigación forense de redes informáticas y los retos a los que el analista se enfrenta antes es necesario tener en cuenta la diferencia entre dos tipos fundamentales de información: datos en reposo y datos en tránsito. De ella dependen tanto los métodos de trabajo como las normativas legales aplicadas y el tipo especial de precauciones que el investigador deberá desplegar cuando trabaje con redes.

Datos en reposo son los que se encuentran almacenados en los ordenadores y pueden ser recuperados siguiendo los métodos descritos a lo largo de este libro. Los elementos de evidencia se encuentran allí: tan solo hay que buscarlos y aplicar procedimientos que aseguren la cadena de custodia. Por el contrario los datos en tránsito son aquellos que tienen una existencia efímera mientras circulan de unos dispositivos a otros a través de la red. Ejemplo típico son las contraseñas, que transitan la red pero no quedan –o no deberían quedar– almacenadas en las máquinas. Los datos en tránsito no dan más que dos opciones al investigador: examinar archivos de registro en busca de pruebas de su existencia o interceptarlos mediante un *sniffer* instalado en un nodo de la red.

En el caso de los archivos de registro –como por ejemplo el syslog de Linux, los eventos de Windows, *logs* de enruteadores, firewalls e IDS–, se pueden hacer copias de los mismos, calcular *hashes* o dotarlos de una firma digital y documentar sus características (tamaño, procedencia, marcas de tiempo, etc.). Todos estos procedimientos son útiles para asegurar la integridad de los datos. La firma digital permitirá mantener una cadena de custodia, a condición de que solo unas pocas personas tengan acceso a la clave. Una falta de claridad en estos procedimientos puede ser aprovechada por la parte contraria para impugnar la evidencia. El esfuerzo de documentación debe ser igualmente diáfano y previsor con los datos obtenidos mediante programas de escucha como Wireshark. Todo esto implica una inversión de tiempo considerable, pero facilita la trazabilidad de aspectos importantes en el enorme volumen de datos que es necesario procesar durante una investigación.

Los sistemas de red también pueden contener información crucial en un estado intermedio entre los dos a los que se ha hecho mención con anterioridad: ni en reposo ni en tránsito, sino estacionados provisionalmente en las memorias RAM de dispositivos como *routers*, puntos de acceso inalámbricos o los propios ordenadores. Estos datos se perderían en caso de apagar el sistema o desconectar el cable de red. Un ejemplo de estos datos provisionalmente retenidos en la RAM serían las conexiones abiertas listadas por netstat, sobre todo cuando indican la IP remota desde la cual se lleva a cabo la intrusión en el sistema. En un capítulo anterior se ha hablado de métodos especiales para hacer volcados de RAM y preservar la información volátil. También existen procedimientos similares para trasladar a un archivo la memoria de un *router* o cualesquiera otros dispositivos de red.

### 6.1.4 Siguiendo pistas

Las dificultades de una investigación en red, considerables de por sí, se ven incrementadas cuando surge la necesidad de ir más allá de los confines de la LAN. Entonces ya no se tratará de un analista examinando imágenes forenses con un ordenador y sus herramientas habituales en la quietud del laboratorio, sino de un especialista que se ve obligado a moverse en un escenario complejo e imprevisible en el que intervienen más personas, y donde es necesario trabajar en equipo y cumplir trámites legales y burocráticos de todo tipo sin que los resultados estén en ningún momento asegurados.

¿Se acuerda de la ejecutiva del primer capítulo, aquella que trabaja para un contratista del sector de la defensa y escuchaba con perplejidad los ruidos causados por la entrada de la policía en una lonja de ciberdelincuentes? ¿La misma a la que una mujer de la limpieza le roba información del ordenador con un Live-CD de Linux y una llave USB? Hace años hubo otro caso de intrusiones en la empresa. En aquella ocasión la prensa incluso llegó a publicar algunas noticias relativas al caso. Un intruso se las había arreglado para instalar un *rootkit* en tres de los cinco servidores del centro de cálculo. El software malicioso fue descubierto por casualidad por uno de los administradores tras observar algunas discrepancias de formato en los listados de procesos activos.

Un examen forense de los servidores afectados permitió hallar archivos de registro creados por el *rootkit* que evidenciaban la existencia de una puerta trasera a través de la cual el intruso accedía al sistema. En el archivo de registro figuraba la dirección IP remota del atacante, lo cual permitía seguir su rastro hasta el proveedor de acceso a Internet. Inmediatamente la policía solicitó una orden judicial y exigió al proveedor de acceso que le entregara los datos correspondientes a aquellas conexiones. Simultáneamente el administrador de la red local comenzó a monitorizar el tráfico dentro del segmento de red utilizado por el intruso para acceder al sistema.

En el curso de la investigación pudo comprobar que los accesos se producían desde una dirección IP fija correspondiente a la conexión ADSL contratada por un usuario particular. La policía averiguó quién era el titular del número telefónico, obtuvo una orden de registro y se presentó en el domicilio del presunto intruso, donde procedió a incautar su ordenador junto con varios soportes de datos y cierta cantidad de documentos. En un disco duro portátil se hallaron los archivos de registro de los *sniffers* con nombres de usuarios y contraseñas. Allí también se encontró un archivo comprimido con la extensión “tar.gz” y en su

interior el *rootkit* instalado en los servidores. Finalmente una búsqueda de caracteres en el espacio no asignado del disco duro de un ordenador perteneciente al sospechoso permitió hallar rastros de lo que parecían ser listados de archivos de servidores comprometidos en otras empresas del sector.



Figura 6.2. Documentación hallada en un registro de la Policía (Fuente: *El Mundo TV*)

Todos estos elementos de evidencia eran útiles porque permitieron relacionar a una persona concreta –el dueño del ordenador– con las pruebas de la intrusión halladas en los servidores de la empresa. Normalmente, sin embargo, rastrear a los ciberdelincuentes no resulta tan fácil, y mucho menos demostrar su culpabilidad delante de un tribunal. Quizás algunos de ellos sean todavía tan incautos como para cometer errores de aficionado, pero eso es algo que sucede cada vez con menos frecuencia, al menos en los casos graves. Vaya haciéndose a la idea de que algunos de estos intrusos no son simples gamberros informáticos que aplican recetas aprendidas en algún foro de *hackers*, sino especialistas con conocimientos de alto nivel en tecnologías de la información, como lo demuestra en ocasiones el material hallado durante los registros. No se dejarán atrapar fácilmente.

### 6.2 PROTOCOLOS

Resulta imposible hacer un buen trabajo en la investigación forense de redes sin tener sólidas nociones de la tecnología en que aquellas se basan. Antes de analizar archivos de registro o manejar cualquier herramienta de intercepción de tráfico es necesario entender cómo se mueven los datos de unas máquinas a otras. Los ordenadores no se limitan a ponerlos sin más en el cable o en el aire –en el caso de las transmisiones inalámbricas– sino que lo hacen aplicando de manera

estricta unos protocolos de red. Un protocolo es un conjunto de reglas y estándares que regula la transmisión de información entre dispositivos de red, y que comprende todos los aspectos del proceso, desde la negociación de conexiones entre el origen y el destino hasta el formato interno y el tamaño de los bloques en que ha de ser dividido el flujo de datos.

Supongamos que un usuario quiere visualizar una página web y la pide desde su navegador. Antes de que una sola línea de código HTML aparezca en pantalla el sistema operativo tiene que hacer unas cuantas cosas. En primer lugar toma la URL de la barra de direcciones del navegador y la introduce en la parte de datos de un paquete TCP, al que asigna un puerto de destino –normalmente el 80 para servidores HTTP– y otro de origen para que la respuesta pueda volver a la aplicación que la solicita, además de información adicional y números de secuencia que en el caso de un volumen de datos más grande que hubiera de ser dividido en varios paquetes permitan una reconstrucción completa del archivo en el destino.

El paquete TCP es pasado a la capa de red donde será envuelto literalmente en un nuevo paquete provisto de direcciones IP de origen –la de la máquina del usuario– y de destino –la del servidor de páginas web, previamente obtenida a través de una consulta a un servidor DNS–. Este nuevo paquete constituye la carga de una trama de red que se forma, acto seguido y en una capa de software inferior, a base de añadir una cabecera dotada de otro tipo de direcciones de origen y de destino: números MAC para los interfaces Ethernet. Cada máquina conectada a Internet posee al menos uno de ellos, y las direcciones MAC de 48 bits, asignadas en virtud de un convenio internacional, son únicas y específicas para cada dispositivo Ethernet.

Finalmente la tarjeta de red transforma todo el flujo de datos en una señal convenientemente modulada para poder viajar hacia su destino a través de un cable, ondas hertzianas o cualquier otro medio. Los datos, empaquetados en toda esa estructura de cabeceras anidadas, son enviados a la puerta de enlace por defecto de la red local, normalmente un *router*, y de allí al proveedor de acceso que los introduce en Internet a través de alguna de las tecnologías de redes de área amplia (WAN) existentes. El paquete llega a su destino, la petición se procesa y es enviada la respuesta.

Este recorrido que ha hecho el paquete de datos a través de aplicaciones concurrentes que se lo van pasando unas a otras resume de un modo simplificado el funcionamiento de lo que llaman pila de protocolos. Mediante estas operaciones, que se verifican de un modo totalmente automático, ejecutadas y supervisadas por módulos de software que corren en segundo plano, las tramas de información pueden desplazarse a través de redes, enruteadores, puentes, *switches* ATM, cortafuegos, sistemas de detección de intrusos y otros nodos. Durante el trayecto

cada dispositivo añade y quita a las tramas lo que en cada momento haga falta: direcciones MAC o IP de las máquinas que dentro de cada red sean las encargadas de procesar el tráfico, nuevas cabeceras para tunelar el tráfico o etiquetas que permitan enrutarlo a través de la WAN. Una vez en el destino, otro sistema operativo que funciona con la pila de protocolos estándar realiza el mismo trabajo pero en sentido inverso, pelando las tramas para convertirlas en paquetes IP, extrayendo de estos a su vez los paquetes TCP y tomando finalmente de estos últimos la URL de la página solicitada y un número de puerto para poder cumplir su objetivo inicial de entregarla al servidor HTTP.

Los protocolos están estructurados en capas. El modelo de capas canónico, que se estudia en carreras de Informática y facultades de Ingeniería, es el denominado OSI, con un total de siete capas (física, enlace de datos, red, transporte, sesión, presentación y aplicación). En la práctica lo que se utiliza es el modelo de capas TCP/IP de Internet, compuesto únicamente por cuatro niveles (acceso al medio, red, transporte y aplicación) que cubren todo el espectro funcional de las capas OSI.

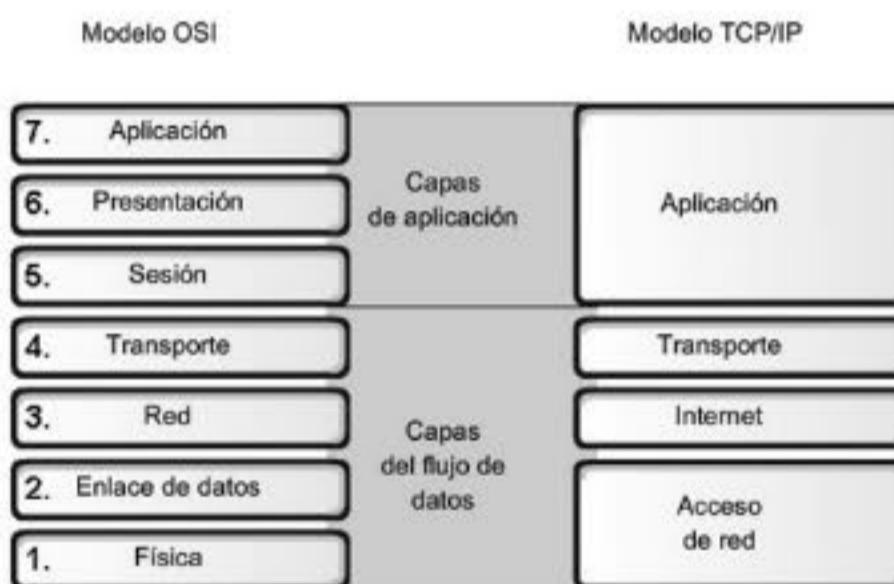


Figura 6.3. Pilas de protocolos OSI y TCP/IP

### 6.2.1 Capa de transporte: TCP

TCP es un protocolo orientado a conexiones que funciona por encima de las capas de red, enlace de datos y física de la pila del modelo OSI, y por debajo de las capas de sesión, presentación y aplicación, las cuales se encargan de procesar datos a más alto nivel. TCP establece circuitos virtuales para determinadas aplicaciones que por motivos de integridad en la transmisión de datos las vayan a necesitar. Tal es el caso de HTTP, correo electrónico, FTP, Kerberos y otras. Para este fin TCP, además de números de puerto que permitan establecer la conexión, introduce códigos de secuencia, confirmación y verificación para hacer posible el

reenvío de paquetes perdidos o recibidos defectuosamente. Existen otros servicios que funcionan en la capa de transporte, como por ejemplo *streaming* de vídeo o VoIP, y no tienen requerimientos de integridad tan elevados. Para estos servicios se dispone de otro protocolo denominado UDP.

Puerto origen 16 bits		Puerto destino 16 bits			
Número de secuencia 32 bits					
Número de confirmación (Acknowledgement) 32 bits					
Longitud cabecera 4 bits	Reservado (6 bits)	U R G A C K P R S H S T Y N F I N N	Tamaño de ventana 16 bits		
Suma de verificación TCP 16 bits		Puntero urgente 16 bits			
Opciones (si las hay)					
Datos (si los hay)					

Figura 6.4. Cabecera TCP

En la figura 6.4 se puede observar la estructura típica de una cabecera TCP. Destacan en primer lugar los números de 16 bits para los puertos de origen y de destino, y después los apartados de 32 bits reservados para los números de secuencia y confirmación. Estos sirven para reensamblar los paquetes en el destino en caso de que la transmisión no sea ordenada por razones técnicas de la red o de cualquier otro tipo. La suma de verificación de 16 bits sirve para comprobar la integridad del paquete. En caso de estar corrupto, TCP solicitará al origen que se lo envíe de nuevo.

## 6.2.2 Puertos

Los puertos son números convencionales a los que el sistema asocia un determinado servicio ejecutado por una aplicación software que quiere comunicarse con otras a través de una red. En el análisis forense conocer estos números resulta útil para detectar software malicioso o conexiones anómalas. Algunos puertos son utilizados legítimamente por el sistema y las aplicaciones del usuario. Si al ejecutar un comando netstat el investigador ve gran cantidad de servicios conectados a puertos con números comprendidos entre 1 y 1.023, como por ejemplo 139 o 445, lo más frecuente es que se trate de una situación enteramente normal. Probablemente el ordenador forma parte de una red en la que se comparten recursos Microsoft a través de NetBIOS. Sin embargo en determinadas circunstancias el estado de estos puertos puede tener interés forense:

un programa escuchando en el puerto 21 da pie a sospechar un intento de intrusión, sobre todo porque casi ningún administrador de sistemas se sirve en la actualidad de una herramienta tan antigua como Telnet para el manejo remoto de ordenadores. Pero algunos ciberdelincuentes sí porque constituye una herramienta útil y una vía muy interesante para desplegar maniobras intrusivas contra ordenadores.

Los puertos comprendidos entre 1.024 y 49.151 son los denominados puertos registrados, y los puede utilizar cualquier aplicación. A partir del puerto 49.152 y hasta el 65.535 se encuentran los llamados puertos dinámicos, que generalmente se asignan a determinadas aplicaciones cliente de manera dinámica al iniciar una conexión. Estos puertos son muy utilizados por programas P2P, por lo que conviene tenerles puesto el ojo encima en todo momento. El investigador puede consultar los documentos de referencia del IANA para obtener información relativa a los puertos de red y las aplicaciones que se sirven de ellos para establecer comunicaciones en red.

Versión 4 bits	Longitud cabecera 4 bits	Tipo de servicio (TOS) 8 bits	Longitud total 16 bits			
			R F	D F	M F	Offset de fragmentación 13 bits
Time To Live (TTL) 8 bits			Protocolo 8 bits			Suma de verificación de la cabecera 16 bits
			IP de origen 32 bits			
			IP de destino 32 bits			
Opciones (si las hay)						
Datos						

Figura 6.5. Cabecera IP

## 6.2.3 Capa de red: IP

En una red funcionan docenas de protocolos. El más conocido de todos ellos sin duda es IP (*Internet Protocol*), ya que sobre él se implementa la estructura básica de todas las redes locales y corporativas que componen esa gran autopista de la información que llamamos Internet. La misión de IP consiste en proveer a los paquetes de datos de todo lo que necesitan para llegar a su destino –generalmente otras máquinas– a través de nodos y enruteadores. En el esquema de la figura 6.5 el lector puede ver cómo es el encabezado de un paquete IP típico, que se añade a los datos procedentes de la capa superior (transporte) con el objeto de proveer direcciones IP de origen y de destino de 32 bits, además de otras

informaciones útiles. Conviene aclarar que estas informaciones para el control de la longitud de los datos, sumas de verificación, etc., no tienen nada que ver con las que vimos al hablar de la capa de transporte y las conexiones TCP. Cada grupo de datos es procesado por su propia capa de software para sus propios y específicos fines.

Una dirección IP es un número que identifica de manera única y jerárquica al interfaz de un dispositivo que forma parte de una red basada en el protocolo IP. Las direcciones IP no se asignan a discreción del usuario ni de los administradores de redes, sino que son adjudicadas por un organismo llamado ICANN (*Internet Corporation for Assigned Names and Numbers*), que a través de un sistema de convenciones unificadas hace posible la existencia de una Internet única en todo el mundo. Esto no quiere decir que en caso de no seguir las convenciones de ICANN no podrían existir redes tipo Internet. Podría haberlas a condición de tener sus propios enrutadores para dirigir el tráfico. Sus paquetes incluso podrían ser transportados por la misma infraestructura de redes de área amplia que soporta la Internet convencional. Pero serían redes cerradas y sus ordenadores solo podrían verse entre sí, sin posibilidad de tener ningún contacto directo con los que forman parte de otras redes. Por consiguiente, y a pesar del problema causado por la escasez de direcciones IP –que en un futuro próximo se espera remediar gracias a la implantación del nuevo protocolo Ipv6–, a todo el mundo le conviene seguir las directrices de ICANN.

Existen muchos tipos de redes conforme a las reglas de ICANN, pero a los propósitos del investigador conviene hacer una distinción entre dos ámbitos muy importantes: la red pública, compuesta por rangos de direcciones asignados por los proveedores de acceso a partir de los grupos de números asignados por ICANN, y la red local, en la que cada administrador puede poner sus propios números dentro de unos rangos considerados de libre disposición según las especificaciones de ICANN. En la categoría de redes de clase C (hasta 254 hosts), que son las más habituales en el mundo de las redes domésticas y la pequeña y mediana empresa, el rango utilizable para las direcciones privadas incluye los números comprendidos entre 192.168.0.0 y 192.168.255.255. Estos números tienen la particularidad de que no son enrutables fuera de la red local. Ningún paquete IP que los lleve incrustados como dirección de destino será capaz de atravesar un router para salir a Internet. La puerta de enlace por defecto siempre los devolverá a la red local, o bien los eliminará en caso de que correspondan a máquinas no existentes.

A viceversa sucede algo similar. Cuando el software de la capa de red detecta un paquete con dirección de destino perteneciente a alguna de las clases reservadas por ICANN para las redes públicas, el router lo dirigirá a otro dispositivo de enrutamiento. Si el paquete es recibido por la puerta de enlace, esta

lo hará llegar al proveedor de acceso de Internet o al dispositivo WAN encargado de procesar la información destinada al exterior. Si no fuera posible hacer nada de esto el paquete será eliminado. Pero en ninguna circunstancia volverá a la red local.

### 6.2.4 Enrutamiento

Por consiguiente los enrutadores no solo sirven para guiar tráfico en las redes, sino que además constituyen una especie de frontera entre Internet y la red local. Antes dijimos que los paquetes con direcciones públicas no son enrutables en la red local. Entonces, ¿cómo hacemos para navegar por Internet sin que nuestro ordenador disponga de una dirección pública expresamente asignada? Cuando el usuario desea ver una página web u obtener recursos de un servidor remoto sus paquetes IP son enviados en primer lugar a una puerta de enlace por defecto –por lo general el *router*– en la que mediante una simple comparación lógica que consiste en aplicar una máscara de subred toma la primera de una serie de importantes decisiones. Si se trata de una dirección privada, lo envía a la red local. Si la dirección es pública, la manda al proveedor de acceso a través del enlace contratado con este: líneas dedicadas, conexión ADSL o cable. Y desde allí la infraestructura mundial de redes de área amplia y otros enrutadores remotos se encargarán de hacer llegar el paquete a su destino.

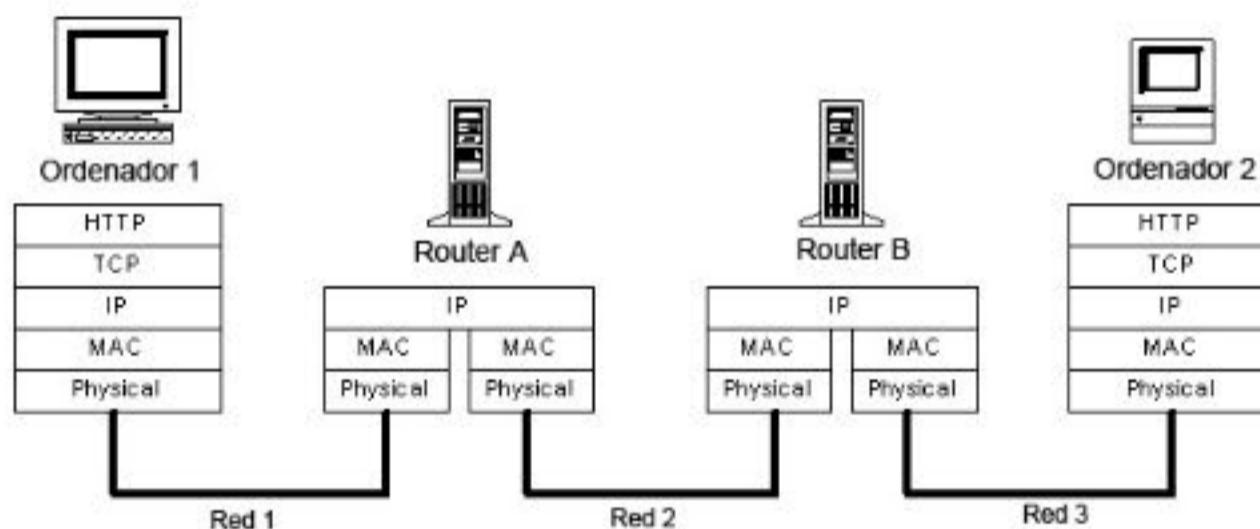


Figura 6.6. Enrutado IP

Tratándose de un componente clave de la red era de esperar que el *router*, además de esta función de guarda fronterizo entre las redes de área amplia (WAN) y las redes de área local (LAN), terminara asumiendo otras competencias de interés forense que justifican todo esfuerzo realizado por el investigador para familiarizarse con la tecnología de estos dispositivos. A diferencia del *router* de hace tan solo algunos años, que solamente ejecutaba tareas básicas de acuerdo con las especificaciones de ICANN, un dispositivo de red moderno no solo separa el tráfico haciendo posible la conexión entre Internet y la red local. También es el

emplazamiento ideal para incorporar funciones de cortafuegos, servidor DHCP, IDS, punto de acceso inalámbrico o WAP, un discriminador para desviar tráfico externo dirigido a servidores situados en la zona desmilitarizada o DMZ y, en el caso de los denominados comutadores multicapa, incluso funciones de filtrado y control mediante análisis de datos pertenecientes a las capas superiores de la pila de protocolos, especialmente la de aplicación. Todo ello combinado en un dispositivo integral eficaz y de gran potencia. Los *routers* constituyen un campo en el que los delincuentes informáticos se ven obligados a trabajar con tesón por razones obvias: para llevar a cabo una intrusión es necesario pasar por ellos, a no ser que ya se esté dentro de la red local o se pueda acceder a ella por medio de un módem telefónico.

Por el mismo motivo los archivos de registro de un *router* poseen gran interés para el investigador forense. Los dispositivos modernos disponen de sistemas automatizados como Netflow capaces de registrar información de cada una de las conexiones que pasa a través de ellos, mandándola al *host* del administrador o almacenándola en la ubicación que aquel indique. El examen de *logs* Netflow puede aportar datos útiles para esclarecer incidentes como instalación de troyanos, espionaje mediante *keyloggers* o intentos de acceso a través de redes virtuales privadas. Siempre será de gran ayuda no solo conocer estas posibilidades, sino también tenerlas configuradas de manera permanente en prevención de posibles anomalías técnicas o vulneraciones de seguridad.

### **6.2.5 Capa de enlace de datos: interfaces Ethernet**

En el transcurso de sus pesquisas el investigador puede llegar a reunir gran cantidad de datos relativos a irregularidades y anomalías en un entorno de red. Acaso los indicios conseguidos o su intuición le hagan sospechar de alguien. Pero, ¿cómo hace para relacionar toda esta información con la persona que ha cometido el delito? Las direcciones IP no son específicas de cada máquina, sino que son asignadas arbitrariamente por el administrador o, caso más frecuente, un servidor DHCP de forma totalmente automática. La suplantación de una IP, ya sea mediante manipulación de paquetes, envenenamiento de cachés ARP o DNS y ataques del tipo *man in the middle*, es tan fácil que ningún juez se tomaría en serio un alegato basado en pruebas procedentes exclusivamente de la capa de red. Una IP no es más que un número que ha sido asignado temporalmente a una máquina. Es más, por sí solas las direcciones IP no resuelven el problema del enrutamiento, ni en redes locales ni en Internet. Aunque un ordenador tenga asignada una dirección IP en la red, por sí mismo no tiene ningún modo de saberlo. Para ello se necesita otro tipo de información. Si el investigador ha entendido la relación que hay entre las

conexiones TCP y las direcciones IP, es hora de que comprenda la que existe entre estas últimas y las direcciones MAC de los interfaces Ethernet.

Ethernet es un estándar de comunicación para redes locales basado en un protocolo de acceso al medio con detección de portadora y colisiones. Como tal, define características de cableado y señalización de nivel físico y formatos de tramas para la capa de enlace de datos del modelo OSI. Esta tecnología, creada por Robert Metcalfe a comienzos de los años 70, aparece implementada en interfaces de conexión para ordenadores y otros dispositivos de red, incluso maquinaria industrial. Actualmente la práctica totalidad de los equipos informáticos – ordenadores de sobremesa, portátiles, routers – vienen provistos de serie con uno o varios interfaces Ethernet. Pero si el lector se ha visto alguna vez en la necesidad de instalar una tarjeta de red en un ordenador ya tiene una idea del tipo de dispositivos que incluyen tecnología Ethernet y para qué se emplean.

*Figura 6.7. MAC de un interfaz Ethernet (“Dirección física”)*

Cada interfaz Ethernet tiene una dirección propia, llamada MAC (*Media Access Control* = Control de Acceso al Medio) y consistente en un número de 48 bits que se asigna en virtud de un convenio internacional. Este número de 48 bits figura agrupado en 12 caracteres hexadecimales que se pueden ver fácilmente listando las características del interfaz de red del ordenador. Por ejemplo en Windows basta abrir una consola de texto (**Ejecutar programa**, cmd.exe) y teclear lo siguiente (figura 6.7):

C:\Users\djal>ipconfig /all

Por supuesto es posible ocultar la dirección MAC o falsearla por medio de utilidades de software que asignan al interfaz Ethernet un número que no le corresponde. Pero esto se puede hacer solo de modo dinámico con el ordenador conectado a la red. En cuanto el sistema se reinicie por cualquier motivo, volverá a estar presente en la red con su MAC original. La dirección MAC está grabada en el hardware literalmente a troquel, y a no ser que otros indicios apunten a la actuación de un ciberdelincuente extremadamente habilidoso y capaz de tener en cuenta detalles de este tipo, por lo general debería bastar para identificar una máquina determinada. En otras palabras la dirección MAC viene a ser como una especie de carné de identidad para un interfaz Ethernet.

Los números MAC constituyen el elemento esencial de una trama de datos Ethernet, que es el receptáculo donde se envuelven los paquetes IP antes de incluirlos en la señal física y enviarlos a la red en forma de impulsos eléctricos u ondas electromagnéticas. Y de este modo hemos conseguido reunir todas las piezas del puzzle: en diferentes niveles de anidamiento, como si se tratara de uno de esos juegos de muñecas rusas que vienen metidas unas dentro de otras, la trama de datos Ethernet transporta todo lo necesario para establecer una comunicación efectiva entre máquinas de origen y de destino: los números MAC que permiten identificar el interfaz Ethernet, las direcciones IP de cada *host* en la red local, el número de puerto para hacer posible la comunicación entre aplicaciones de software concurrentes y además una cantidad considerable de información adicional: números de secuencia, sumas de verificación, señalizadores de prioridad y urgencia, etc. Un protocolo adicional, ARP (*Address Resolution Protocol*) se encarga de establecer la correspondencia entre números MAC fijos y direcciones IP arbitrarias, preguntándoseles a los enrutadores o a las máquinas adyacentes mediante mensajes de difusión o *broadcasting*.

Por debajo de la capa de enlace de datos se encuentra el nivel físico: todo lo relativo a señales eléctricas, ondas electromagnéticas, pulsos luminosos y tecnología hardware de bajo nivel. Por lo general y debido al funcionamiento automático de los procesos digitales que regulan el acceso de los componentes electrónicos al medio de transmisión, no es mucho lo que este nivel tiene que ofrecer al investigador para las finalidades que este persigue.

## 6.2.6 Protocolos de nivel superior: HTTP y SMB

HTTP (*Hyper Text Transfer Protocol* = Protocolo de Transferencia de HiperTexto) y SMB (*Server Message Block* = Bloque de Mensaje de Servidor) son protocolos que funcionan en la capa de aplicación y por consiguiente en un nivel superior a las de transporte, red y enlace de datos. Dejamos para el final aquello

por lo que deberíamos haber comenzado, siguiendo el orden lógico descendente de nuestra exposición. Si se ha hecho así ha sido por razones de claridad y de orden expositivo: en primer lugar interesaba explicar la tecnología básica de la red, con los mecanismos automáticos que permiten trasladar paquetes de datos de unas máquinas a otras, haciendo posible el funcionamiento de las redes de ordenadores, y después la forma en que el usuario, cómodamente instalado por encima de las capas de nivel inferior, se sirve de sus aplicaciones para sus propios fines. HTTP, junto con otros protocolos y programas auxiliares, sirve para crear la experiencia de navegación por Internet que conocemos gracias a los *browsers* de uso extendido como Internet Explorer, Mozilla o Google Chrome, mientras que SMB es el protocolo utilizado por sistemas operativos MS-Windows para compartir archivos, impresoras y otros recursos.

Para un investigador es importante entender la estructura básica de HTTP porque en la actualidad los navegadores de Internet se han convertido no solo en la herramienta principal del usuario para moverse en el ciberspacio, sino también en una aplicación que se emplea para todo tipo de procesos de comunicación, administración y acceso a recursos de red. Aunque HTTP está configurado para utilizar TCP en el puerto 80, el administrador puede configurar su servidor web en cualquier otro puerto. El tráfico HTTP también puede estar encriptado con HTTP sobre TLS (*Transport Layer Security* = Seguridad en Capa de Transporte), funcionalidad que da origen a un protocolo combinado bajo la denominación HTTPS. Este protocolo es muy útil para acceder a servicios donde la confidencialidad es importante, como por ejemplo correo web, portales de comercio electrónico o sitios de Internet de las entidades bancarias. El usuario reconocerá la presencia del protocolo HTTPS por venir indicado en la URL de la barra de conexiones, los avisos que el servidor le envía cada vez que establezca una conexión segura o por la presencia de un pequeño ícono con un candado en la parte superior de la ventana del navegador.

Cada vez que se establece una conexión con un servidor web para acceder a uno de sus recursos (páginas, aplicaciones web o bases de datos) se crea una entrada en un archivo de registro con información detallada acerca de la máquina de origen, la fecha y la hora de la acción y el código devuelto por el servidor. Aunque el formato de estos *logs* es específico de cada servidor, casi todos ellos utilizan el estándar denominado CLF (*Common Log Format*) que admite diversas posibilidades de configuración, incluyendo el navegador y su versión, la cantidad de *bytes* entregados, etc.

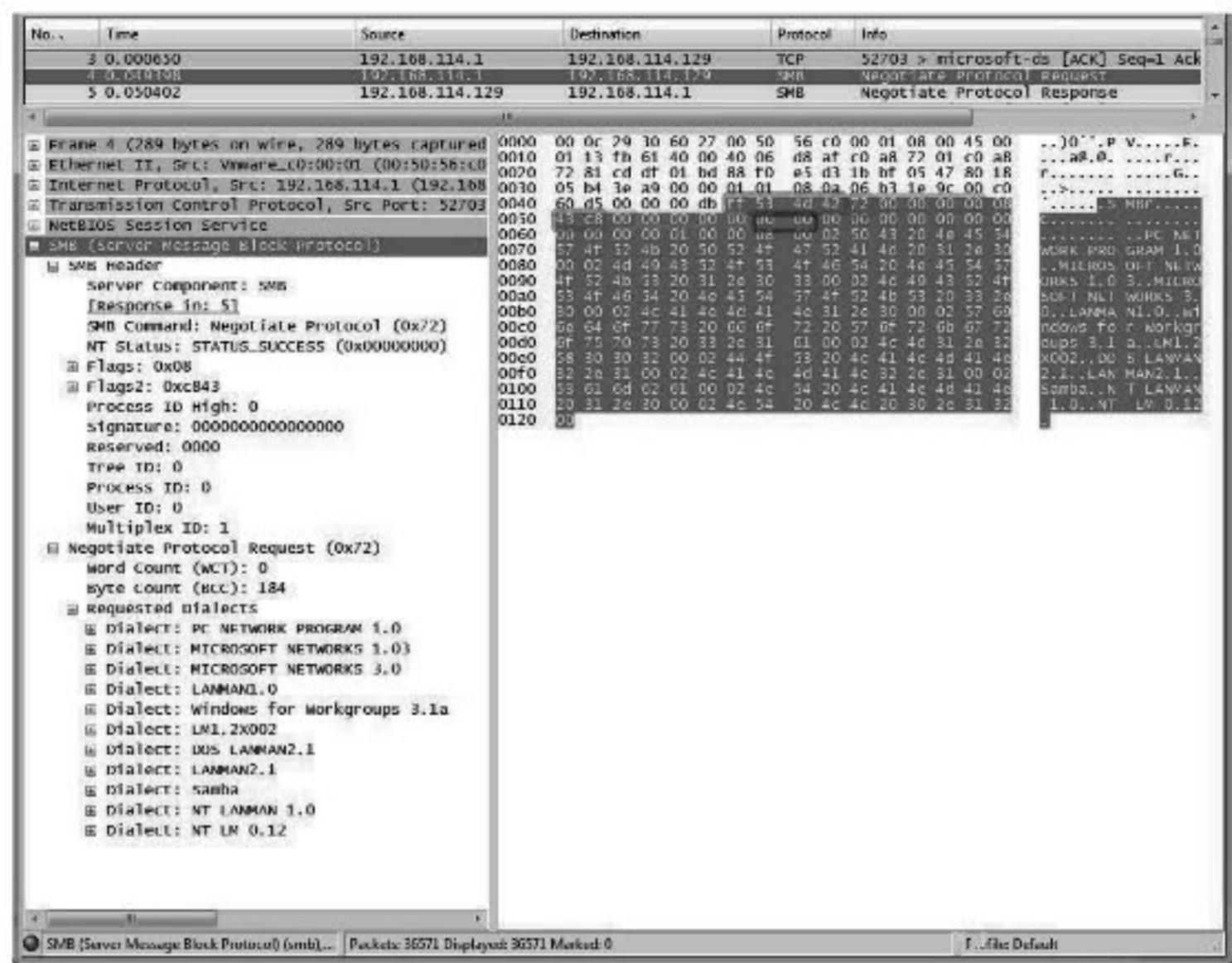


Figura 6.8. Cabecera SMB vista a través de un analizador de tráfico

SMB es un protocolo de la capa de aplicación que utiliza un enfoque típico cliente-servidor en el que el cliente envía peticiones de recursos y el servidor responde de modo conveniente en función de las políticas de acceso establecidas. La conexión se establece por lo general a través de TCP en el puerto 445 –aunque la antigua API de Microsoft NetBIOS, precursora de SMB, utilizaba el puerto 139–. Para recursos que por razones funcionales no sean incompatibles con la tecnología subyacente (sistemas operativos, API, etc.) también se admiten peticiones procedentes de Linux/Unix a través del software SAMBA. SMB dispone de numerosos comandos y el paquete de Linux correspondiente incluye información significativa sobre gran cantidad de aspectos de configuración y relativos a las prestaciones del software. El investigador puede extraer de una sesión SMB informaciones de indudable utilidad forense como por ejemplo el nombre del usuario que la abre y los recursos a los que aquél accede. Para ello no necesita más que analizar la relación entre los identificadores de proceso (PID), multiplexado (MID), usuario (UID) y árbol (TID) en la cabecera del paquete

(figura 6.8). UID y PID son evidentes; MID permite seguir la pista a las diferentes peticiones realizadas por un mismo proceso, y TID identifica conexiones con las diversas cuotas de usuario (*shares*) tras el establecimiento de una conexión.

## 6.3 ANALIZANDO EL TRÁFICO DE RED

Estará preguntándose el lector cuál es el verdadero significado práctico, sobre todo a efectos de una investigación forense, de todo lo que se le está contando, y si no sería posible dejarse de tanta teoría sobre protocolos, modelos de capas y paquetes para pasar sin más a las aplicaciones concretas y al modo de emplearlas para obtener un resultado inmediato. Con lo aprendido hasta el momento resultará fácil entender el funcionamiento de aplicaciones como Tcpdump o Wireshark utilizadas para monitorizar tráfico de redes. Básicamente lo que hacen las herramientas de este tipo es interceptar paquetes en interfaces Ethernet o inalámbricos y mostrar su contenido al investigador. Los paquetes aparecen desglosados por capas, tipos de protocolo y otros parámetros.

El analizador de tráfico o *sniffer* es un instrumento apreciado por *hackers* y administradores de red que también puede rendir un valioso servicio al investigador forense. Pero para sacarle partido antes era necesario saber cómo funcionan las redes informáticas. Abraham Lincoln dijo una vez que si le dieran seis horas para cortar un árbol dedicaría cuatro a afilar concienzudamente el hacha y las dos restantes al trabajo de brazo propiamente dicho. Del mismo modo, si el lector tuviera que convertirse en un experto en redes y un año para hacerlo, debería emplear nueve meses en estudiar a fondo los protocolos de red y ocuparse durante los tres restantes de todo lo demás: hardware, sistemas operativos, utilidades de administración, etc.

### 6.3.1 Wireshark

Wireshark, antiguamente llamado Ethereal, es un analizador de protocolos de código libre cuya misión consiste en facilitar el estudio del tráfico en tiempo real en redes de ordenadores al efecto de localizar problemas que entorpecen el rendimiento de las comunicaciones corporativas. Por el mismo motivo también constituye una herramienta excelente para el aprendizaje y la investigación forense. A diferencia de otros programas diseñados para la misma finalidad y que funcionan en línea de comando, Wireshark dispone de un cómodo y práctico pero no por ello menos potente interfaz gráfico. En la actualidad existen versiones de Wireshark para Linux y Windows.

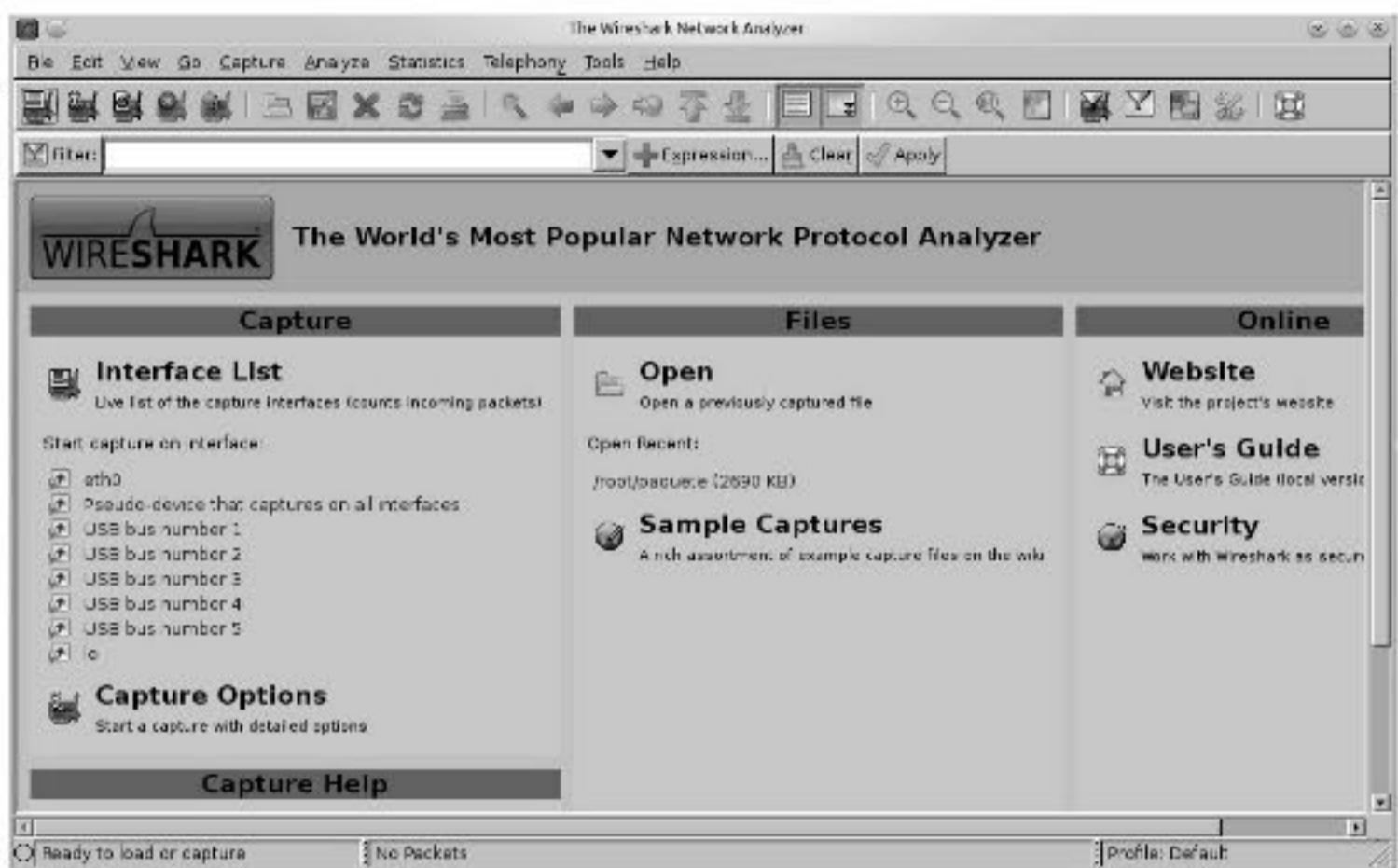


Figura 6.9. Ventana inicial de Wireshark

Wireshark dispone de gran cantidad de filtros para afinar el estudio de los paquetes interceptados y es capaz de reconocer más de 1.000 protocolos en el flujo de datos de la red. El diseño de su interfaz de usuario es diáfano y sencillo: una parte superior, situada bajo la barra de menús, para definir y configurar filtros, que permite establecer patrones de búsqueda específicos a fin de poder seleccionar con ellos los paquetes y protocolos que interesa interceptar. Acto seguido, y en la parte central de la ventana, se puede ver la lista secuencial de visualización de todos los paquetes atrapados en tiempo real. Una correcta interpretación de los datos de esta sección permitirá al investigador deducir cuál es el problema que afecta al funcionamiento de la red o, en su caso, hallar elementos de evidencia para su investigación forense. Más abajo encontramos un recuadro de visualización en el que tras haber resaltado el paquete que nos interesa, marcándolo con el ratón en la sección anterior, podremos examinar su contenido y todos los protocolos anidados que contiene mediante un sistema de menús desplegables. Finalmente, en la parte inferior de la ventana un visor hexadecimal muestra el contenido detallado de los datos del paquete capturado por el interfaz de red.

### 6.3.2 Captura de tráfico: hubs, mirroring, bridges

Supongamos que acabamos de recibir una alerta de seguridad y tenemos que monitorizar el tráfico de una red corporativa para ver si se están produciendo intrusiones de algún tipo o un uso indebido de la infraestructura informática de la

empresa. Nos gustaría hacer algunas pruebas con Wireshark dentro de la LAN. ¿Por dónde empezamos? Si tenemos localizado el destino de los intentos de acceso, por ejemplo un servidor de documentos o recursos para la Intranet, la respuesta es evidente: entre el servidor y el *router* más próximo. La primera cuestión, por lo tanto es: ¿dónde conectamos nuestro portátil con nuestro sistema forense Windows o Linux y Wireshark ejecutándose sobre él? No podremos hacerlo directamente y sin más al *router* que comunica al servidor con el resto de la red, porque en la actualidad la mayor parte de los dispositivos de enrutamiento son conmutadores o *switches*, y dirigen el tráfico de un modo en función de los interfaces Ethernet a los que va destinado, sirviéndose para ello de una lógica especial y tablas de direcciones MAC/IP. Dicho de otra manera, nuestra estación de trabajo forense únicamente podría captar los paquetes dirigidos a ella desde el *switch* al que nos hayamos conectado, pero no los paquetes que el *switch* transporta entre el servidor y otras máquinas de la red.

Una posible solución consiste en insertar un concentrador o *hub* entre el servidor y el *switch*. Nuestro ordenador portátil con Wireshark iría entonces conectado a cualquiera de las bocas restantes disponibles del *hub*. El *hub* funciona como un repetidor que recibe una señal de datos por uno de sus puertos y la difunde por los otros con la suposición de que el *host* al que va dirigida la recogerá mientras que las restantes máquinas ignorarán sin más los paquetes. Fue el primer sistema de interconexión de redes de ordenadores mediante nodos, pero no tardó en ser sustituido por los *switches* porque estos, gracias a su lógica de control, permitían la creación de dominios de colisión separados en la red, haciendo de este modo posible un mejor aprovechamiento del ancho de banda. Al haber caído en desuso es difícil que podamos disponer de un *hub* en el lugar de la investigación, a no ser que por azar encontremos uno en algún cuarto trastero o llevemos uno de nuestra propiedad. Por otra parte, si prevemos que nos vamos a dedicar con frecuencia a la monitorización de redes, tampoco estaría de más conseguir un *hub* de segunda mano comprándolo en eBay o en cualquier almacén de productos informáticos desechados.



Figura 6.10. Hub Ethernet

Si no disponemos de un *hub* existe la alternativa de captar el tráfico desde el mismo *switch* mediante *mirroring* de puertos, siempre que el dispositivo de commutación admita esta funcionalidad. El *mirroring* permite duplicar el tráfico que circula por uno o varios puertos del *switch* haciéndolo salir por otro que hayamos configurado al efecto y al cual irá conectada nuestra estación de trabajo. Este método lo emplean los administradores de sistemas para instalar IDS y otras herramientas de supervisión de red. No olvide que el puerto por el que está escuchando el tráfico tiene que ser tan rápido o más que aquel cuyo tráfico tiene la intención de interceptar, ya que de lo contrario Wireshark perdería parte de los paquetes.

Otras posibilidades, en el supuesto de que el *switch* no admita *mirroring* ni se disponga de un concentrador, pasan por diversos montajes poco prácticos pero que en determinadas situaciones podrían servir, como por ejemplo la instalación de un *bridge* o puente implementado a base de un ordenador con dos interfaces de red y un software especial como bridge-utils (Linux) para hacer pasar el tráfico de uno a otro, capturándolo con Wireshark en cualquiera de ellos. Como último recurso podríamos intentar una maniobra muy popular entre *hackers* pero extremadamente ruidosa y contaminante: el ARP *Spoofing*. Consiste en envenenar la caché ARP mediante paquetes de aviso falsificados, para que el *switch* desvíe el tráfico a nuestra estación de trabajo, devolviéndolo acto seguido en dirección a sus destinos originarios. Esto crearía una especie de intermediario virtual entre el servidor y el *switch*, pero para mantenerlo sería necesario un bombardeo constante de la caché con datos falsos a fin de evitar que esta se limpie durante un ciclo rutinario de actualización. En cualquier caso se trata de un método intrusivo que no se podría utilizar en entornos críticos, y menos si hay un IDS instalado en la red.

### 6.3.3 Utilización de Wireshark

Wireshark está disponible para descarga en <http://www.wireshark.org/>, donde además de documentación y otros recursos el lector encontrará versiones para diferentes arquitecturas de hardware –Intel 32 y 64 bits– y sistemas operativos –Windows y OSX–. Si desea la versión para Linux podrá obtenerla en <http://wireshark.softonic.com/linux> o instalarla directamente desde un repositorio. En ambos casos necesitará las librerías Pcap para captura de paquetes. El instalador de Windows las lleva empaquetadas en su interior. De no ser ese el caso las podrá encontrar en <http://www.winpcap.org>. En Linux tendrá que conseguir el paquete correspondiente. Ubuntu resuelve las dependencias e instala las librerías automáticamente. Para ello no tiene más que teclear:

```
sudo apt-get install wireshark
```

El interfaz de Wireshark, ya explicado en un apartado anterior, no tiene mayor misterio. En las páginas que siguen se utiliza la versión para Linux, pero en Windows todo viene a hacerse prácticamente de la misma manera. Al ejecutar el programa lo primero que el usuario ve es la ventana principal con las cuatro partes anteriormente mencionadas.

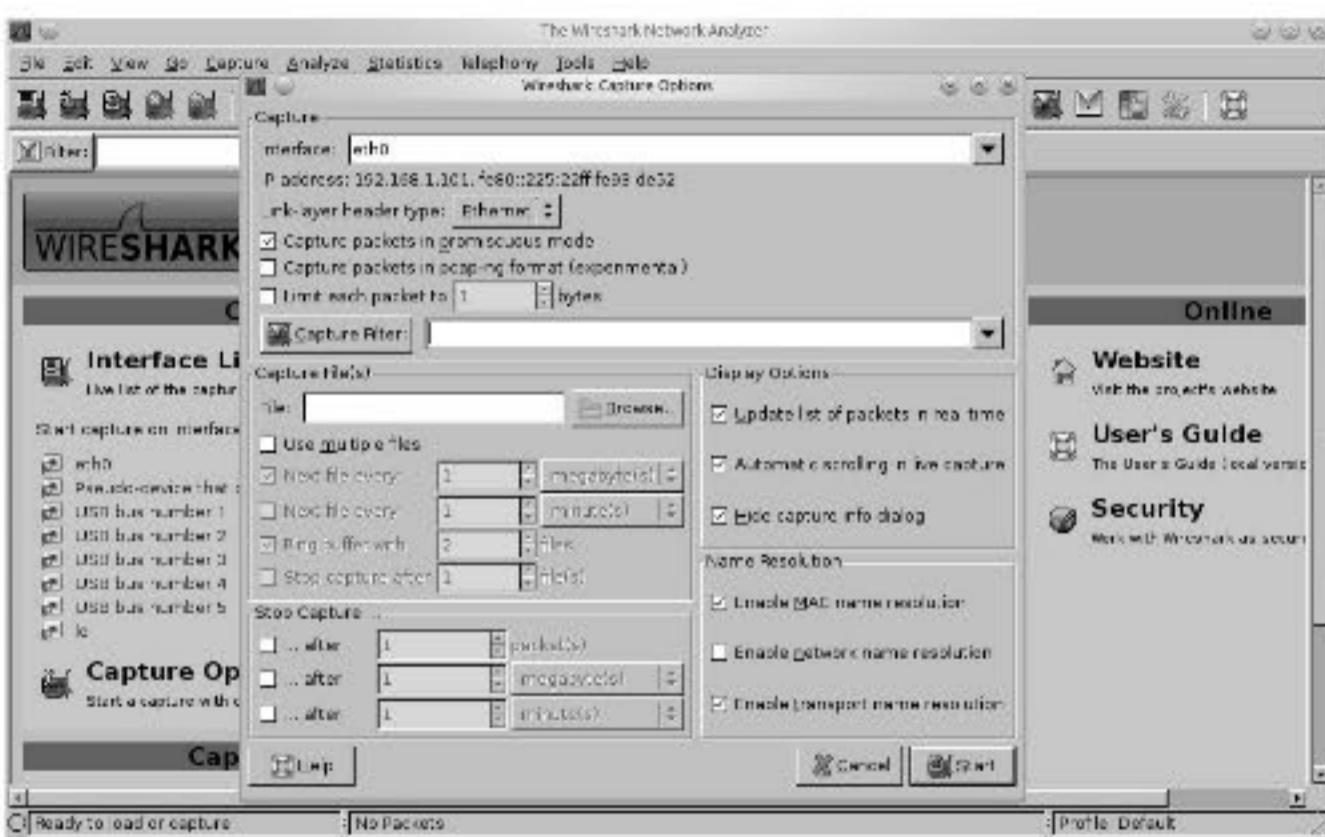


Figura 6.11. Wireshark – Opciones de configuración

Para comenzar la captura de paquetes con Wireshark lo único que tiene que hacer es ir en primer lugar al menú de configuración (**Capture → Options**), en el que además de comprobar si su interfaz de red ha sido detectado correctamente podrá ajustar algunos parámetros de funcionamiento (figura 6.11). La captura de paquetes comienza después de haber seleccionado la interfaz correspondiente (**Capture → Interfaces**). Si nuestra estación de trabajo está conectada de alguno de los modos explicados en el apartado anterior, basta con pulsar el botón correspondiente a la interfaz local Ethernet. Para realizar su trabajo, esta es commutada automáticamente por Wireshark a un modo especial de funcionamiento que se denomina “promiscuo”. Este no es el modo que el interfaz Ethernet utiliza durante su servicio normal para atender los requerimientos de comunicación del usuario. En el modo normal una tarjeta de red verifica las direcciones MAC de las tramas de datos para ver si van dirigidas a ella. En caso contrario las tramas son ignoradas y eliminadas en la misma capa de enlace de datos. Sin embargo en modo promiscuo el interfaz de red deja subir a las capas superiores (red, transporte y aplicación) todas las tramas, para que Wireshark y otras utilidades de monitorización de redes puedan analizar las cabeceras de los paquetes y los datos transportados por los mismos.

Muy pronto tendrá ocasión de ver cómo la ventana se llena de paquetes. El resultado de la sesión puede guardarse en archivos de formato pcap que después podrán ser estudiados con Wireshark o con otras herramientas. Para afinar la investigación y evitar volúmenes excesivos de datos Wireshark ofrece al usuario la posibilidad de configurar filtros de acuerdo con una amplia variedad de criterios. Por ejemplo podrá limitar la captura a los elementos relacionados con una dirección IP, con un protocolo determinado o con elementos de búsqueda concretos, de modo similar a las búsquedas de caracteres en el análisis de una imagen forense. También podrá obtener resúmenes estadísticos del tráfico (Statistics), aplicar la heurística propia del programa (Analyze → Expert Infos) para obtener pistas de lo que está haciendo un usuario de red o extraer del flujo de información contenidos de diversos tipos (File → Export → Objects). Otras posibilidades de configuración de las muchas que Wireshark ofrece al usuario consisten en personalizar la presentación de los datos mediante colores para facilitar el seguimiento de conversaciones, utilidades para telefonía IP, etc.

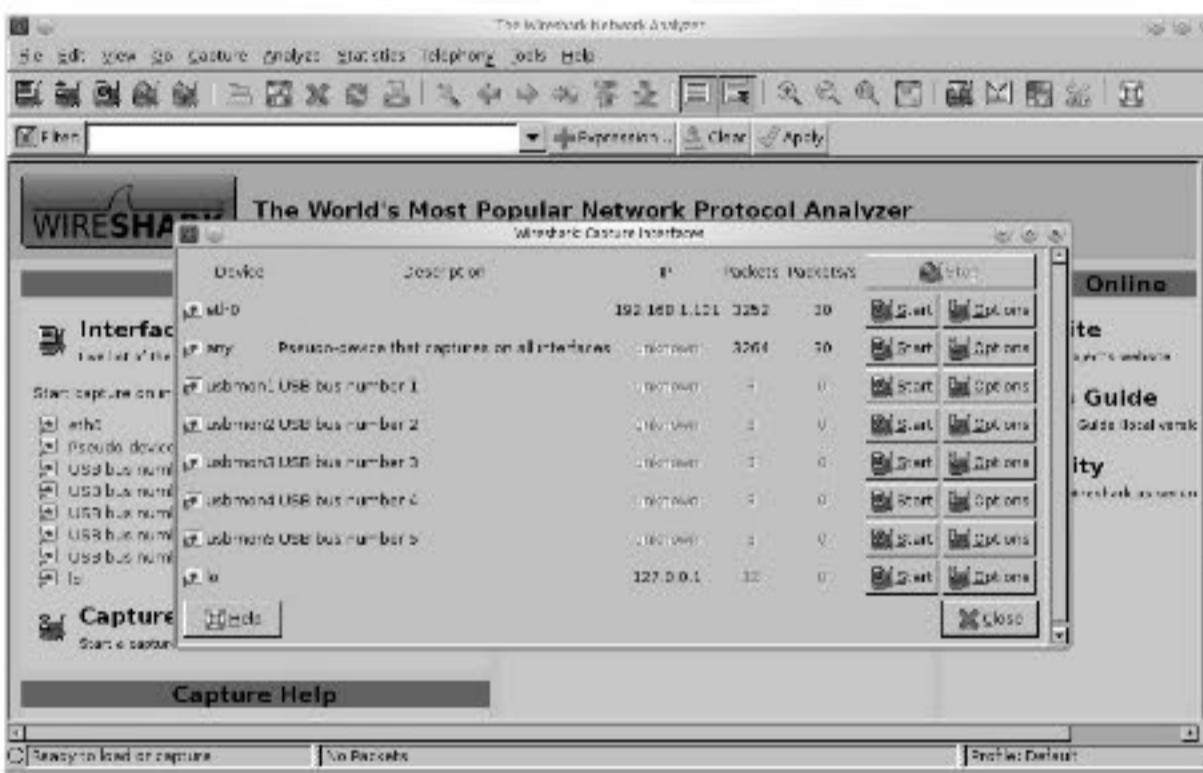


Figura 6.12. Wireshark – Selección de un interfaz para la captura

Wireshark también puede ser utilizado para monitorizar redes inalámbricas. En tal caso deberá tener en cuenta un número de detalles técnicos que es necesario conocer para la intercepción de paquetes en un medio electromagnético, como por ejemplo la pluralidad de canales en los que emiten los puntos de acceso, los modos específicos de funcionamiento de los interfaces WiFi (*managed, monitor, ad hoc*), posibles fuentes de interferencia, etc. Así mismo hay que tener en cuenta que la transmisión inalámbrica utiliza protocolos adicionales con los que el investigador deberá familiarizarse. En el ejemplo de la figura 6.13 teníamos nuestra estación forense, un portátil con Linux Backtrack 5, a la escucha en una red pública sin seguridad. En la ventana de Wireshark se puede ver un

dispositivo de marca Apple conectado. Podría tratarse de un MacBook, un iPhone o un ordenador de sobremesa con tarjeta inalámbrica en algún domicilio de la vecindad. En cualquier caso Wireshark hace posible la identificación de la máquina facilitando la dirección MAC de su interfaz WiFi.

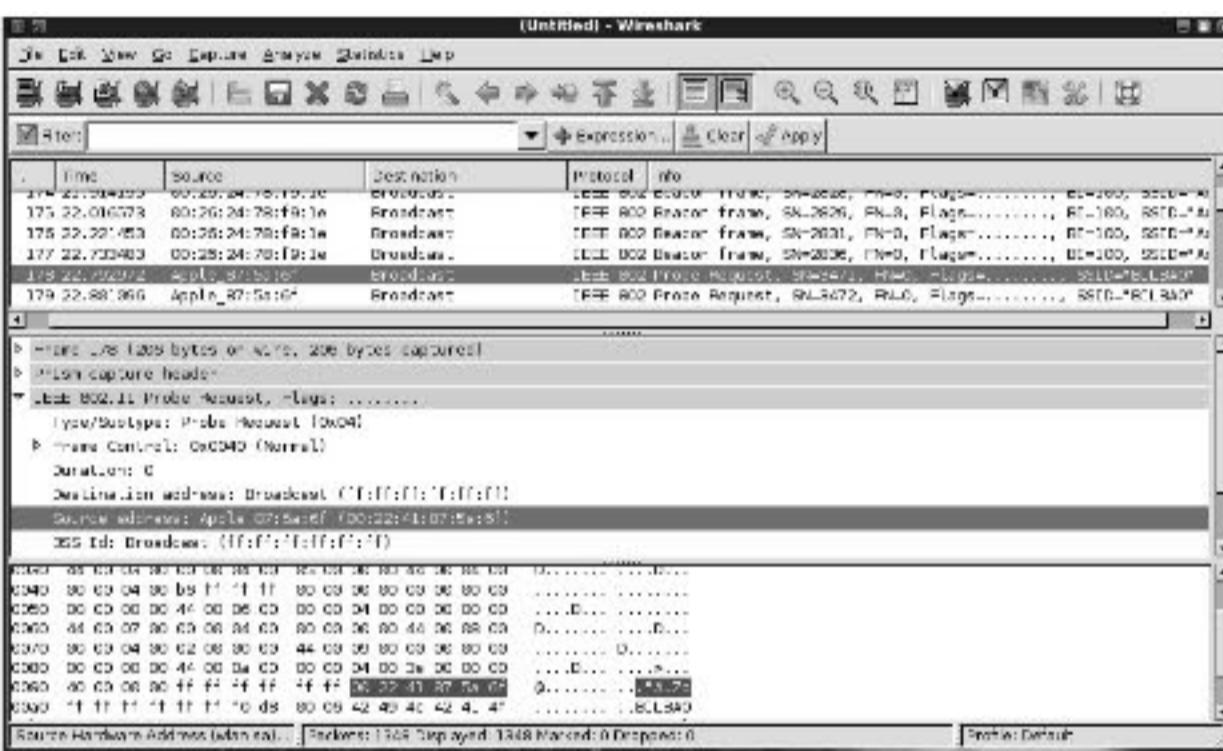


Figura 6.13. Captura de una trama WiFi (protocolo IEEE 802.11) por Wireshark

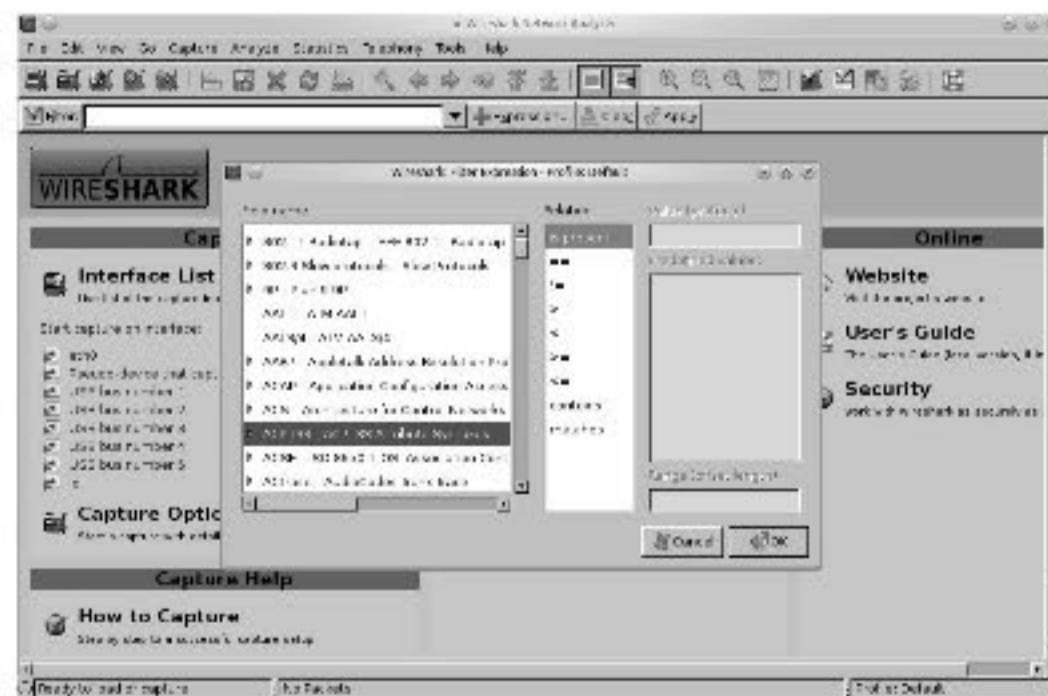
Tampoco hace falta decir que, al igual que en el caso de la intercepción de paquetes en una red Ethernet con cableado físico, para la escucha inalámbrica también se requieren equipo y preparativos adicionales: una buena tarjeta de red —preferiblemente USB tipo Alfa o similar— y antenas. Finalmente no olvide que servirse de Wireshark en un entorno inalámbrico no es lo mismo que utilizarlo en una red cableada. Windows requiere controladores especiales para acceder al modo de funcionamiento monitor de las tarjetas inalámbricas. En Linux lo más aconsejable es utilizar una distribución especial para auditoría como Backtrack, que además de Wireshark ya instalado lleva consigo todo lo necesario para que un interfaz WiFi funcione a pleno rendimiento, aparte de gran cantidad de herramientas utilizables para fines forenses (visualizadores de particiones, dd, md5sum, TSK, etc.).

### 6.3.4 Un ejemplo práctico

Suponga que el propietario de una red corporativa le pide que lleve a cabo una investigación para evaluar el uso de la infraestructura con vistas a la elaboración de un manual de buenas prácticas para el personal de la empresa. Recientemente se han detectado algunos casos de infección de troyanos. Alguno de los ordenadores de la red podría estar siendo utilizado como repositorio de *warez* o pornografía. Quizás con el tiempo esta empresa se ha convertido en un nido de *bots*.

y ahora sin darse cuenta en realidad trabaja para la Russian Business Network, aportando su granito de arena en ataques de denegación de servicio contra servidores de grandes corporaciones, los gobiernos de algunos países o el FBI. Todos sabemos cómo se llega a tal situación. El administrador de la red sospecha que los usuarios, por mucho que aseguren que no navegan por sitios raros, no están haciendo un uso de los recursos informáticos de la empresa razonable y estrictamente acorde a los requerimientos de sus puestos de trabajo.

El lugar indicado para conectar su estación de trabajo forense es el *router* que actúa como pasarela entre la red local y el proveedor de acceso a Internet. Afortunadamente se trata de un dispositivo Cisco de última generación y alto rendimiento que admite *mirroring* de puertos, por lo que desviar el tráfico hacia un portátil no es cuestión más que de configurar una opción en el dispositivo. Acto seguido inicia Wireshark y comienza a captar tráfico aleatoriamente. Como estamos a las mismas puertas de Internet y por ahí pasa todo el tráfico de la red local, el volumen de datos será considerable. Es posible incluso que el interfaz de su estación no pueda captarlo todo. Afortunadamente no es preciso quedarse con todos los paquetes. Usted sabe que el principal vector de infección está constituido por lo que los usuarios se bajan de páginas web inseguras. Eso es lo que hay que filtrar: métodos GET y POST de peticiones HTTP. Confecciona una sentencia adecuada en el apartado de filtros de Wireshark y la pone en funcionamiento (figura 6.14).



*Figura 6.14. Configurando filtros con Wireshark*

Al cabo de un tiempo detiene la captura, guarda el archivo y dedica un rato a examinar su contenido, en busca de pistas que le permitan afinar la búsqueda mediante filtros adicionales. Hay, en efecto, algunas direcciones IP de la red local de las cuales partieron, durante la hora del almuerzo, peticiones HTTP un tanto

inusuales. Por consiguiente, el ciclo de vigilancia posterior estará centrado en este objetivo. Ahora el caudal de paquetes ha disminuido lo suficiente como para dejar su estación de trabajo en funcionamiento durante un buen rato sin temor a que el volumen del archivo resultante lo haga inmanejable. Al final de la jornada detiene Wireshark, desconecta su estación de trabajo del *switch* y hace inventario. Redacta un breve informe, lo saca por la impresora más próxima y se lo traslada al administrador de la red y al gerente de la empresa para que lo examinen. Como prueba lleva también su propio portátil y les muestra los elementos de evidencia, un tanto perturbadores, rescatados gracias a File → Export → Objects (figura 6.15).

Afortunadamente no hemos hallado rastros de intrusiones ni de aplicaciones maliciosas que pudieran estar comunicándose con un interlocutor remoto. Por el momento parece que esta empresa se encuentra a salvo de posibles daños de imagen o complicaciones jurídicas derivadas de la responsabilidad civil por daños o del incumplimiento de la ley de protección de datos. Tampoco cabe culpar a los empleados por un mal cuyo origen se encuentra no solo en la conducta negligente de aquellos sino también en la ausencia de un manual de procedimientos adecuados y buenas prácticas. Simplemente ha llegado la hora de prevenir males mayores mediante el establecimiento de una política de uso de Internet basada en el sentido común y en los resultados de nuestro informe. Antes de poner manos a la obra el gerente de la oficina, impresionado por la capacidad de Wireshark, le pide a usted que prolongue su observación durante dos o tres días, para de este modo disponer de una base de conocimiento más amplia acerca del problema.

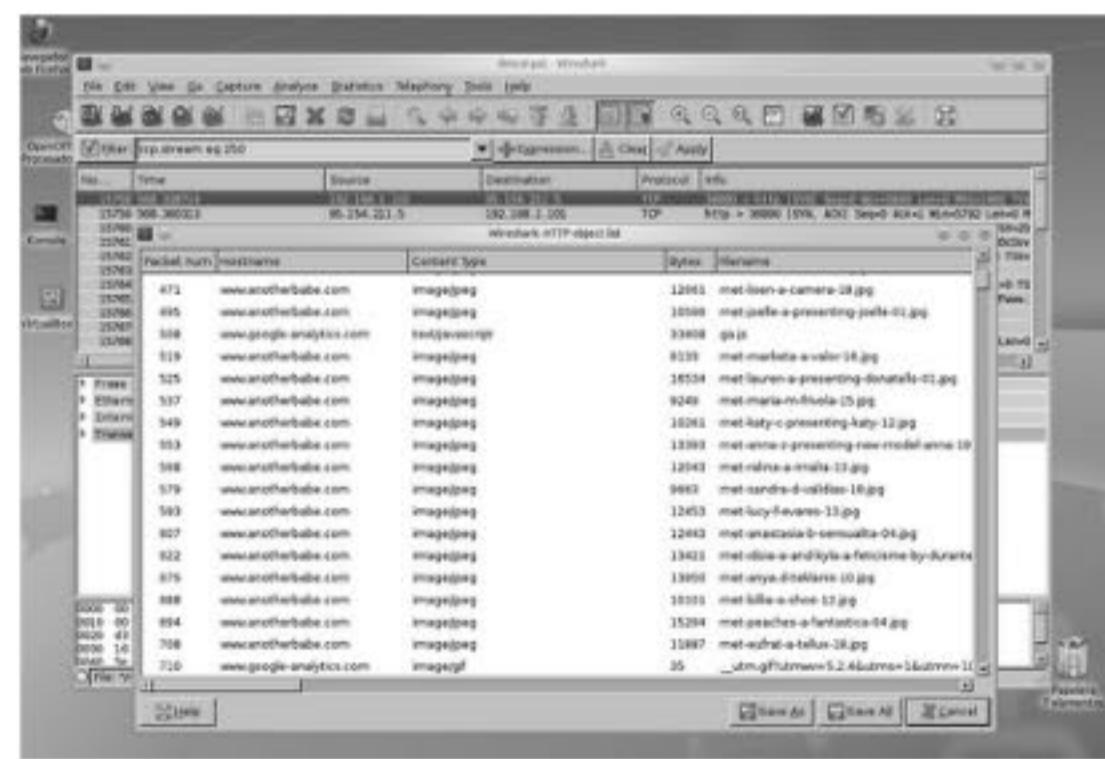


Figura 6.15. Evidencia de uso indebido de la red obtenida con Wireshark

## 6.4 COMPROBACIÓN DE DIRECCIONES IP

### 6.4.1 Herramientas de traza de red

Internet es un espacio público con grandes oportunidades para el anonimato, pero no tantas como habitualmente se piensa. Podemos encriptar nuestros datos o alojarlos en servidores extranjeros fuera del alcance de la ley, pero en general todo está perfectamente localizado. Algo por lo demás necesario, ya que los ordenadores necesitan un sistema de referencias únicas y bien identificadas para funcionar cuando se conectan unos a otros. Diversas herramientas como ping, whois, traceroute y las incluidas en el paquete BIND (*host*, *dig* y *nslookup*), desarrolladas en los comienzos de la red para facilitar la búsqueda de información en un parque de *hosts* que crecía a un ritmo exponencial, nos permiten orientarnos en Internet de modo parecido a un astrónomo con una carta estelar. Nombre, posición y características de los cuerpos celestes del ciberespacio –*hosts*, servidores y *routers*– se hallan al alcance de nuestros dedos con solo abrir una consola *bash* y teclear los comandos adecuados. Estas utilidades dependen de la tolerancia de *routers* y servidores a su actuación y no siempre funcionarán, pero en la mayor parte de los casos le resultarán útiles al investigador como herramientas de análisis no solo en Internet, sino también –en el caso de ping y traceroute– en redes locales e intranets.

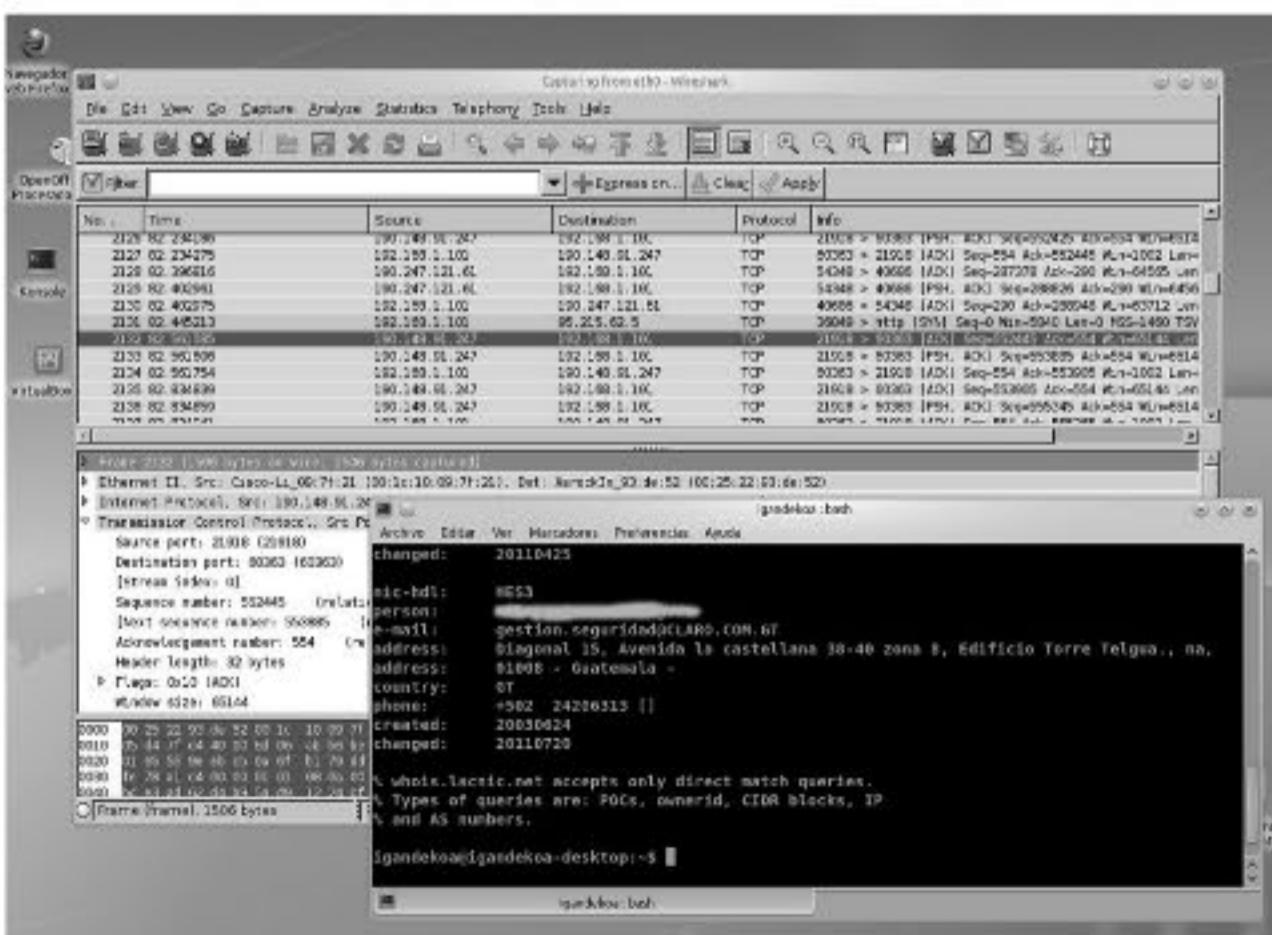


Figura 6.16. Consulta Whois sobre una dirección IP hallada con Wireshark

### 6.4.2 Whois o quién es quién en Internet

*Whois* y *fwhois* son herramientas en línea de comando que nos permiten consultar bases de datos especiales con el objeto de obtener información sobre nombres de dominio y direcciones IP. Los servidores whois están gestionados por autoridades de nombre de dominio de todo el mundo y contienen diversos tipos de información. Por lo general lo que más nos interesa es la ubicación, datos de contacto y rango de direcciones IP que se encuentran bajo su autoridad. Whois suele venir instalado por defecto en la mayoría de las distribuciones Linux. Su manejo es harto simple:

```
xim@dasecon:~$ whois inmobiliariaXYZ-asi-construyen.com
```

Cuando no se trata de dominios del tipo COM, ORG o NET, podemos probar con otro de los servidores whois además de los que el programa utiliza por defecto (*whois.internic.net* y *whois.crsnic.net*). Existe un número de ellos especializados en distintos tipos de dominios o zonas geográficas. Lamentablemente España aún no dispone de un servidor whois para nombres de dominio del tipo .ES.

Siguiendo con el ejemplo que acabamos de poner, he aquí parte de lo que podría ser la salida típica para una consulta whois:

```
Registrar: FastDomain Inc.
Provider Name....: YellowHost.Com
Provider Whois...: whois.yellowhost.com
Provider Homepage: http://www.yellowhost.com/
Domain Name: INMOBILIARIAX-ASI-CONSTRUYEN.COM
Created on.....: 2009-10-05 07:50:20 GMT
Expires on.....: 2011-10-05 07:50:20 GMT
Last modified on.: 2010-10-06 08:15:56 GMT

Registrant Info: (FAST-12785240)
Attn: inmobiliariax-asi-construyen.com 1,500 GB Space and 15,000 Monthly Bandwidth.
YellowHost.Com INC
165 Schwabing 2 North
** FREE DOMAIN REGISTRATION **
Hosting plans starting at ONLY $6.95 per month -
Bremerton, Utah 84606
United States
Phone: +1.8017556412
Fax..: +1.8017552300
Email: whois@yellowhost.com
Last modified: 2010-09-13 15:22:54 GMT
```

Whois entrega una cantidad considerable de información, aunque gran parte de la misma consiste en presentaciones, normas sobre el funcionamiento de servidores whois, advertencias sobre descargo de responsabilidad civil y demás. En este caso podemos conseguir detalles haciendo otra consulta whois al nombre de dominio del proveedor de alojamiento:

```
xim@dasecon:~$ whois yellowhost.com
```

Y el resultado (así mismo extractado):

```
OrgName: Yellowhost Inc.
OrgId: YELLOWH-2
Address: 65 Schwabing 2 North
City: Bremerton
StateProv: UT
PostalCode: 84606
Country: US
RegDate: 2006-08-08
Updated: 2009-07-13
Ref: http://whois.arin.net/rest/org/BLUEH-2

OrgTechHandle: RAYMO4-ARIN
OrgTechName: Meyers, Steven
OrgTechPhone: +1-801-755-6412
OrgTechEmail: smeyers@yellowhost.com
OrgTechRef: http://whois.arin.net/rest/poc/STMEO4-ARIN

OrgTechHandle: SAL72-ARIN
OrgTechName: Margolis, Jan
OrgTechPhone: +1-801-755-6412
OrgTechEmail: netops@yellowhost.com
OrgTechRef: http://whois.arin.net/rest/poc/JMAR72-ARIN

RTechHandle: NETWO2081-ARIN
RTechName: Network Operations
RTechPhone: +1-801-755-6416
RTechEmail: netops@yellowhost.com
RTechRef: http://whois.arin.net/rest/poc/NETWO2081-ARIN

RAbuseHandle: NOC2320-ARIN
RAbuseName: Network Operations Center
RAbusePhone: +1-801-755-6412
RAbuseEmail: abuse@yellowhost.com
RAbuseRef: http://whois.arin.net/rest/poc/NOC2320-ARIN

RNOCHandle: TECHN497-ARIN
RNOCName: Technical Operations
RNOCPhone: +1-801-755-6412
RNOCEmail: support@yellowhost.com
RNOCRef: http://whois.arin.net/rest/poc/TECHN497-ARIN
```

Con todos estos datos la policía puede ponerse en contacto con el proveedor de acceso para pedirle que conserve sus archivos de registro a la espera de una orden judicial. En casos de injurias o conflictos jurídicos por vulneraciones de la propiedad intelectual, los abogados de la empresa afectada pueden hacer lo mismo para hacer llegar sus demandas o propuestas de acuerdo, o solicitar el cese de prácticas abusivas, etc.

Tenga en cuenta que la información aportada por Whois no tiene por qué ser verídica. Este es el caso, por ejemplo, cuando las direcciones IP han sido falseadas o el sitio del cual parte el ataque está alojado en proveedores de acceso controlados por organizaciones delictivas como la Russian Business Network. Aunque se trate de un sitio legítimo con señas verídicas también hay que contar con dificultades de acceso o falta de jurisdicción, sobre todo si el proveedor de acceso o alojamientos web tiene su sede en el extranjero. En este caso la empresa que sea víctima de la agresión y esté defendiendo sus derechos deberá ponerse en contacto con los abogados del proveedor para pedirles que paralicen la actividad ilícita, solicitar órdenes judiciales a tal efecto o, en caso de tratarse de delitos graves, realizar denuncias ante la policía y poner el asunto en manos de la administración de justicia. Pero eso ya son otros procedimientos de los cuales no podemos tratar aquí.

### 6.4.3 Ping/fping

Entre la utilería de reconocimiento de redes más elemental se encuentra *ping*, cuya función consiste en enviar paquetes ICMP (Protocolo de Mensajes para Control de Internet) a un *host* de destino y esperar la respuesta procedente del mismo. *Ping* se emplea para verificar la conectividad de la red. Aunque pueda parecer un comando trivial, admite una variedad de usos creativos de gran interés para administradores de sistemas y *hackers*. Manipulando el tamaño de los paquetes ICMP se puede provocar un desbordamiento en la capacidad de la pila de protocolos para manejar datos durante la operación de reensamblado de paquetes, lo cual permite bloquear un *host* de destino dejándolo fuera de línea. Este *bug* o deficiencia de software, que constituye el fundamento de una antigua modalidad de ataque denominada “silbido de la muerte”, no es un problema propio de ping, sino del modo en que IP vuelve a ensamblar en el destino los paquetes fragmentados. Desde hace años resulta prácticamente imposible enviar paquetes ping de tamaño superior al permitido, pero aún se puede hacer con sistemas antiguos como Microsoft NT.

Otro uso atípico de esta herramienta consiste en la posibilidad de emplear direcciones de difusión para saber de modo rápido y sencillo qué equipos de una red están activos. Por ejemplo, si estamos en una red privada utilizariamos una

dirección de difusión del tipo 192.168.1.255. El inconveniente es que con esta técnica también se pueden llevar a cabo ataques de denegación de servicio, introduciendo como IP de origen de los paquetes ICMP la dirección del servidor que se quiere atacar y lanzando pings contra la dirección de difusión de una red grande.

El investigador puede servirse de *ping* para conocer de manera trivial la dirección IP de un dominio de Internet. Una consulta whois a la IP indicada arroja más o menos el mismo tipo de información que cuando se lanza contra nombres de dominio.

Con respecto a Fping o “ping rápido” (*fast ping*), se trata de una utilidad gratuita que no viene instalada por defecto. Podemos encontrarla en repositorios o la página web del desarrollador, <http://fping.sourceforge.net>. Lo que hace es automatizar el funcionamiento de ping. Fping envía paquetes ICMP de manera asíncrona a un número de equipos, esperando respuestas y registrando los resultados. Fping no es de esos comandos con los cuales uno debería dedicarse a experimentar alegremente. No lo utilice a no ser que tenga una buena razón y sepa lo que está haciendo.

#### 6.4.4 Traceroute/tracert

Traceroute es un programa de consola que viene instalado por defecto en Windows –donde hay que teclear “tracert”, pero no en Linux, a no ser que se trate de una distribución de seguridad como Backtrack o para reparación de sistemas como Knoppix o SystemRescueCD. Podemos encontrarlo en cualquier repositorio. Si estamos en Ubuntu, por ejemplo, su instalación resulta sencilla:

```
xim@dasecon:~$ sudo apt-get install traceroute
```

Con traceroute se puede añadir contexto a una investigación determinando el trayecto de redes y nodos por los que pasa una conexión antes de alcanzar su destino. Un listado típico de traceroute, en el contexto del ejemplo precedente, podría ser este:

```
traceroute to inmobiliariax-asi-construyen.com (74.220.xxx.69), 30 hops max, 60 byte
packets
1 mygateway1.ar7 (192.168.1.1) 2.095 ms 3.137 ms 3.958 ms
2 10.6.177.129 (10.6.177.129) 43.703 ms 44.912 ms 46.139 ms
3 241.Red (80.58.122.241) 47.631 ms 49.027 ms 50.276 ms
4 So7 (84.16.8.125) 67.651 ms 68.771 ms 69.680 ms
5 So2 (213.140.36.122) 101.165 ms 102.301 ms 104.230 ms
6 So5-3-0-0-grtnycpt3.red.telefonica-wholesale.net (213.140.38.221) 174.085 ms so6
(213.140.38.157) 159.625 ms So1 (213.140.36.206) 166.837 ms
```

```
7 Xe6 (213.140.37.13) 173.282 ms Xe7 (94.142.124.66) 158.009 ms Xe6 (213.140.37.13)
160.690 ms
8 Xe8-0-0-0-grtwaseq5.red.telefonica-wholesale.net.122.142.94.in-addr.arpa
(94.142.122.185) 162.807 ms Xe2-1-0-0-grtwaseq5.red.telefonica-
wholesale.net (84.16.14.217) 158.473 ms xe10 (213.140.36.61) 162.262 ms
9 XO (213.140.55.78) 235.331 ms 230.168 ms 231.622 ms
10 te (207.88.12.201) 230.660 ms 234.102 ms 237.998 ms
11 vb6.rar3.chicago (207.88.12.33) 242.715 ms 244.028 ms 251.522 ms
12 te (207.88.12.22) 244.382 ms 252.632 ms 238.857 ms
13 ae0d0.mcr1.saltlake (216.156.0.2) 241.755 ms 235.304 ms 235.272 ms
14 ae1d0.mcr1.saltlake2 (216.156.1.2) 227.761 ms 225.862 ms 227.655 ms
15 ip65 (65.46.63.10) 232.594 ms 232.758 ms 233.171 ms
16 Inmox (74.220.xxx.69) 242.636 ms 244.165 ms 251.512 ms
```

*Traceroute* no solo nos permite explorar la geopolítica de Internet, sino que además constituye un valioso instrumento para evaluar el estado de las redes y localizar problemas de comunicación. Funciona mediante envíos sucesivos de paquetes IP (por defecto UDP, e ICMP si lo especificamos mediante el operador -I) en los que el campo TTL de la cabecera IP se incrementa de manera progresiva. TTL (*Time To Live*) indica el número de saltos que se permite dar al paquete antes de que sea eliminado, momento en que el *router* que pone el contador a cero enviará al *host* de origen un mensaje de aviso con su dirección IP y otros datos relativos a su identidad y localización. El primer paquete lleva un TTL=1 y expira al llegar al primer nodo de enlace –por lo general un *router* o la puerta de enlace predeterminada de la red local–. A continuación se emite otro paquete con TTL=2, y así sucesivamente hasta llegar a la máquina de destino. El número de saltos predeterminado es de 30, pero se puede indicar otro. La mayoría de los sistemas consideran las emisiones traceroute como tráfico válido, por lo que lo único que se puede hacer para frenarlos es usar cortafuegos o bien mantenerlos bajo vigilancia mediante sistemas de detección de intrusos. Esta es la razón por la cual a veces en la salida del comando se reciben asteriscos o códigos extraños.

#### 6.5 CORREO ELECTRÓNICO

El correo electrónico puede ser analizado, además de por sus contenidos, en función de determinadas informaciones que figuran incluidas en sus cabeceras. Estas vendrían a constituir una especie de sobre digital en el que figuran destinatario, remitente y diversos sellos de conformidad y control correspondientes a las sucesivas estafetas u oficinas de distribución por las que ha pasado la carta. No conviene olvidar que todos estos apartados son configurables y que las indicaciones que se muestran en los encabezados de correo electrónico no tienen por qué corresponder a la realidad. El análisis del correo, además de tener importancia en casos de espionaje corporativo y deslealtad, ha adquirido

protagonismo ante el creciente auge del *spam* y el envío de mensajes no solicitados.

### 6.5.1 Cabeceras e-mail

El análisis de un mensaje de correo electrónico se lleva a cabo por lo general de acuerdo con el procedimiento siguiente:

- El investigador deberá examinar minuciosamente la totalidad de las cabeceras del correo electrónico.
- Atención al contenido de todas las etiquetas “Received”: la que está más arriba es insertada por el último MTA (*Mail Transfer Agent* = Agente de Correo Electrónico) que transportó el mensaje hasta el buzón de nuestro correo a través de una cadena de servidores. El MTA situado en el extremo inferior (último “Received”) es, o debería ser, el primero en la cadena de transporte.
- Una consulta whois de la dirección IP que figura como origen del mensaje puede revelarnos a qué organización pertenece.
- Finalmente el contacto con los responsables de dicha organización permitiría averiguar la identidad del usuario que tenía asignada esa IP en el momento de enviar el mensaje.

### 6.5.2 Estructura típica de un encabezado

Este ejemplo permite explicar con claridad las diferentes partes de un encabezado típico de correo electrónico:

```
Return-Path: <conserje@asesoriatarmac.com>
Received: from IMPmx1.adm.correo (10.20.102.44) by tems5.backend.correo
          (8.5.113) (Microsoft Exchange Internet Mail Service Version
          5.5.2650.21)
          id 4B34A9C901CB672B for patxi_lazaro@telefonica.net; Wed, 17 Feb 2010
          12:29:20 +0100
Received: from proxypop2.sarenet.es ([194.30.0.95])
          by IMPmx1.adm.correo with BIZ IMP
          id izVG1d00K22yuGa01zVGh4; Wed, 17 Feb 2010 12:29:20 +0100
Received: from LOLA (unknown [212.81.199.51])
          by proxypop2.sarenet.es (Postfix) with SMTP id 46BB6735DD
          for <patxi_lazaro@telefonica.net>; Wed, 17 Feb 2010 12:29:12 +0100
          (CET)
Message-ID: <004701caafc4$4ba13f70$340194c0@tarmac.local>
From: "LABORAL" <laboral@asesoriatarmac.com>
```

```
To: patxi_lazaro@telefonica.net
From: <conserje@asesoriatarmac.com>
Subject: presupuesto auditoría
Date: Wed, 17 Feb 2010 12:28:14 +0100
MIME-Version: 1.0
Content-Type: multipart/mixed;
           boundary="----=_NextPart_000_0043_01CAAFCC.AD3855E0"
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 6.00.2900.2180
Disposition-Notification-To: "LABORAL" <laboral@asesoriatarmac.com>
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.2180
```

“Return-Path” no tiene por qué significar necesariamente el verdadero origen de la comunicación: simplemente quiere decir que al emisor le gustaría que todo el correo de respuesta fuera dirigido a este destinatario. Con frecuencia se trata de una dirección falsa, que en realidad no existe como tal o podría pertenecer a una persona cualquiera con el propósito de desviar hacia ella la investigación. Figura siempre al comienzo del encabezado. En el primer “Received” encontramos el servidor de correo de nuestro propio proveedor de acceso a Internet. Esta es la última estafeta que se ha hecho cargo del mensaje entrante antes de entregárnoslo. Obsérvese que en las líneas del encabezado aparecen diversas indicaciones de fecha y hora. La exactitud de las mismas no está asegurada, sino que depende de la configuración de los diferentes servidores de correo electrónico por los que ha ido pasando el mensaje. También hay que tener cuidado con las direcciones IP y verificar que pertenecen a rangos permitidos por IANA (la autoridad encargada de asignar números de Internet). En caso contrario se trataría de direcciones falsificadas. Así mismo hay que tener en cuenta que existe la posibilidad de incluir líneas “Received” adicionales con información falsa –técnica utilizada habitualmente por los *spammers*– para dificultar el rastreo.

“To:”, después de la cadena de “Received” y el código de identificación del mensaje indica el destinatario de este, si bien de un modo relativo. Para dirigirlo el sistema se sirve de una opción de correo llamada BCC (*Blind Carbon Copy*) que permite mandar un mensaje a un destinatario que no figura indicado en el encabezado. Por lo que respecta al identificador del mensaje (“Message-Id”) se trata de un código irrepetible que hace las funciones de número de serie. Lo inserta el propio sistema de correo saliente del remitente, y en caso de que no sea así, cualquiera de los MTA sucesivos.

Los encabezados del correo electrónico, debido a la complejidad de los sistemas de correo saliente y a la libertad de configuración de las cabeceras, constituye un campo demasiado amplio para los propósitos de esta obra, en el que lamentablemente no nos resulta posible profundizar. Si el investigador necesita más información sobre análisis *e-mail*, *spam* y otras materias relacionadas, puede recurrir al sitio de Internet <http://abuse.net/index.html> y consultar una extensa base de datos con materiales y recursos.

## INVESTIGACIÓN FORENSE DE DISPOSITIVOS MÓVILES

### Capítulo 7

Resulta interesante comprobar cómo a lo largo de los últimos años los dispositivos móviles han experimentado una evolución que presenta numerosos puntos de contacto con la del ordenador personal. La historia de la computación se remonta a proyectos militares e industriales de gran envergadura, como la bomba atómica y los sistemas de reserva de billetes para las compañías aéreas. De modo similar, teléfonos móviles, agendas electrónicas y GPS tienen su origen en las necesidades tácticas de las Fuerzas Armadas o, según se comenta también, en los teclados que los astronautas llevaban integrados en las mangas de sus trajes presurizados. Después de saltar a los mercados de consumo y al gran público, lo mismo que el PC en los años 80 del siglo pasado, la informática móvil vive en estos momentos una especie de explosión cámbrica de propuestas de diseño, sistemas operativos y funcionalidades en la que sin embargo se perfilan algunas soluciones predominantes.

Del mismo modo que el PC quedó más o menos estandarizado en tres variantes principales de software –MS-Windows, GNU/Linux y Apple OSX– sobre unas pocas CPU –Intel, Motorola y, en menor medida, Alpha–, en el momento de escribir estas líneas (comienzos de 2012) el mercado de dispositivos móviles parece estar entrando en una etapa caracterizada por la competencia entre un número reducido de marcas –Apple iOS, Android, Windows Mobile y Blackberry– que funcionan sobre hardware diferente con un número de funcionalidades comunes: conectividad a redes, sonido y cámaras de vídeo, GPS, acelerómetro y giroscopio integrados. Si alguna de estas posibilidades ha de desplazar a las otras es algo que dirá el tiempo, pero por el momento todo apunta a

la coexistencia más o menos pacífica entre dos o tres grandes –o más propiamente dicho pequeñas– plataformas.

Si este libro se hubiera escrito hace un par de años, el capítulo que ahora nos ocupa habría sido diferente: hablaríamos de agendas electrónicas, del Compaq iPaq, del legendario Palm Pilot y de Windows CE. Esto no quiere decir que el investigador ocasionalmente no reciba en su laboratorio la visita de una de esas entrañables antiguallas, que le planteará no pocos problemas a la hora de rescatar datos de su interior. El análisis forense de dispositivos móviles constituye un campo enormemente complejo. Para comprobarlo basta mirar cualquiera de los abundantes libros escritos sobre el tema en el mercado anglosajón. Por razones de conveniencia, en las páginas que siguen nos vemos obligados a dedicar una atención preferente a los dispositivos más utilizados en la actualidad: por un lado el Apple iPhone, incluyendo en el mismo apartado los iPad con pantalla de gran formato que funcionan con el mismo software, y por otro los teléfonos inteligentes y tabletas de diversas marcas basadas en el sistema operativo de código libre Android.

Puesto que las funcionalidades básicas de unos y otros vienen a ser las mismas –acceso a redes de telefonía y de datos, GPS, ejecución de aplicaciones de software, entretenimiento, etc.– en primer lugar hablaremos de las características comunes, sin olvidar cuestiones relativas a la precaria seguridad de datos en los dispositivos móviles. Posteriormente se examinarán de forma específica los diferentes métodos de análisis forense aplicados a los dispositivos móviles Apple y Android. Para terminar tocaremos algunas de las dificultades que el análisis de teléfonos móviles y *smartphones* plantea al investigador y a la defensa legal a la hora de presentar sus resultados y hacerlos valer ante los tribunales, así como diversos aspectos relacionados con la privacidad del usuario y los riesgos legales de la informática móvil.

### 7.1 TELÉFONOS MÓVILES INTELIGENTES

#### 7.1.1 Smartphones: pasaporte al siglo XXI

El más sofisticado de los no tan antiguos, pero cada vez menos utilizados, teléfonos móviles no se puede comparar ni de lejos a la experiencia de poseer un *smartphone* con pantalla táctil de gran tamaño, cámara de vídeo, GPS y posibilidades de acceso prácticamente ilimitadas a redes GSM, WiFi y Bluetooth. En los comienzos de la telefonía móvil estar conectado generaba en el usuario una sensación de seguridad y libertad. Lo han podido comprobar viajantes de comercio, excursionistas, geólogos y familias expuestas al peligro de secuestro en las grandes metrópolis de Latinoamérica. Es cierto que por todo ello había que pagar un precio,

y hay que seguirlo pagando en la actualidad: los *smartphones* –teléfono, ordenador, agenda electrónica, reproductor de medios y geolocalizador todo en uno–, heredan las servidumbres de la telefonía móvil (servicio insatisfactorio, tarifas abusivas, bloqueo de dispositivos). Sin embargo la percepción subjetiva no es la misma debido a un número de ventajas cualitativamente superiores e infinitamente más atractivas para el consumidor. En un iPhone, Samsung Galaxy o HTC, el usuario literalmente toca los datos con los dedos sobre una pantalla que pese a su reducido tamaño comienza a tener la definición de un televisor HD. Si el teléfono móvil supuso la culminación de toda la tecnología de telecomunicaciones del siglo XX, los nuevos *smartphones*, compactos, manejables, adictivos y estéticamente rompedores, son tarjeta obligada de presentación en la aldea global del siglo XXI.

Al igual que su predecesor el PC, el teléfono inteligente es tan transparente en su relación con el usuario como opaco en cuanto a sus interioridades tecnológicas. Intuitividad y frescura en el manejo seducen a las masas sin transmitir una idea adecuada de lo que sucede tras la pantalla de estos dispositivos de apenas diez milímetros de espesor. Subsistemas miniaturizados de gran complejidad trabajan sincrónicamente bajo las órdenes de un sistema operativo iOS o Android. Un potente gestor de bases de datos (SQLite) recopila datos y los va introduciendo en tablas: números de teléfono, contactos, direcciones de correo electrónico, mensajes SMS, llamadas entrantes y salientes, archivos de audio, fotografías, anotaciones en el calendario e incluso posiciones geográficas registradas por el GPS. Cuando las opciones de geolocalización están activadas, un iPhone o Samsung Galaxy no solo saben dónde se encuentra su propietario sino también por dónde ha andado y lo que ha estado haciendo. Recuerda los lugares en los que se detuvo para hacer fotos y aquellos otros en los que el adaptador inalámbrico estableció contacto automáticamente con puntos de acceso WiFi. Al contrario que en las novelas de Marcel Proust, la recuperación del camino y el tiempo perdidos no es sentimental y líricamente difusa, sino objetiva y certera dentro de un margen de metros y centésimas de segundo. Por si fuera poco el usuario está obligado a saber que ya no es dueño exclusivo de la información que él y su aparato generan. Ahora está en poder de Apple, Google y varias compañías de marketing por Internet. Así que puede ir preparándose para recibir publicidad.

Si el lector trabaja para una empresa importante y tiene acceso a datos sensibles no está de más reflexionar sobre las consecuencias que puede acarrearle la pérdida de un *smartphone*. El terminal acumula de manera automática gran cantidad de información que escapa al control directo del usuario y resulta difícil de proteger mediante cifrado. La substracción de un iPhone tiene los mismos efectos que el robo de un portátil o una estación de trabajo: daños de imagen, pérdida de competitividad, responsabilidades civiles, riesgo de intrusión en la red y brechas en la seguridad corporativa.

### 7.1.2 Hardware

El iPhone 4 o el Samsung Galaxy S que la compañía telefónica nos acaba de regalar por puntos, con procesador Apple A4 a 800 MH o ARM Cortex A8 de 1 Gigahertzio, 512 MB de RAM, 16 GB de memoria *flash* –en el iPhone: 32 GB– (equivalente a discos duros de la misma capacidad), tiene una potencia muy superior a la de un ordenador de sobremesa de finales del siglo XX. Aquellas máquinas podían gestionar ya con soltura la pila de protocolos TCP/IP y conexiones de banda ancha a Internet. Los dispositivos móviles van más lejos. No necesitan ser conectados a ningún módem pues vienen provistos de serie con interfaces inalámbricos. Tampoco precisan de teclados ni periféricos de entrada porque se manejan con un solo dedo. Indudablemente el aparato que ha marcado la pauta en los mercados, pese a numerosos experimentos y propuestas predecesoras, es el Apple iPhone, que no solo definió un grupo de funcionalidades básicas y equipamientos de serie –coexistencia de procesadores especializados en comunicaciones con CPU de corte clásico, empleo de bases de datos SQLite, etc.– sino que también marcó la pauta en cuanto al aspecto visual, el tamaño y la estética de los dispositivos. Y lo ha hecho con tanto éxito que hoy día cuesta distinguir de lejos un iPhone de algunos de los artículos de la competencia basados en Android. Estos dispositivos nuevos están quitando cuota de mercado al iPhone, ya que ofrecen a precios más económicos prestaciones iguales o en algunos casos superiores.

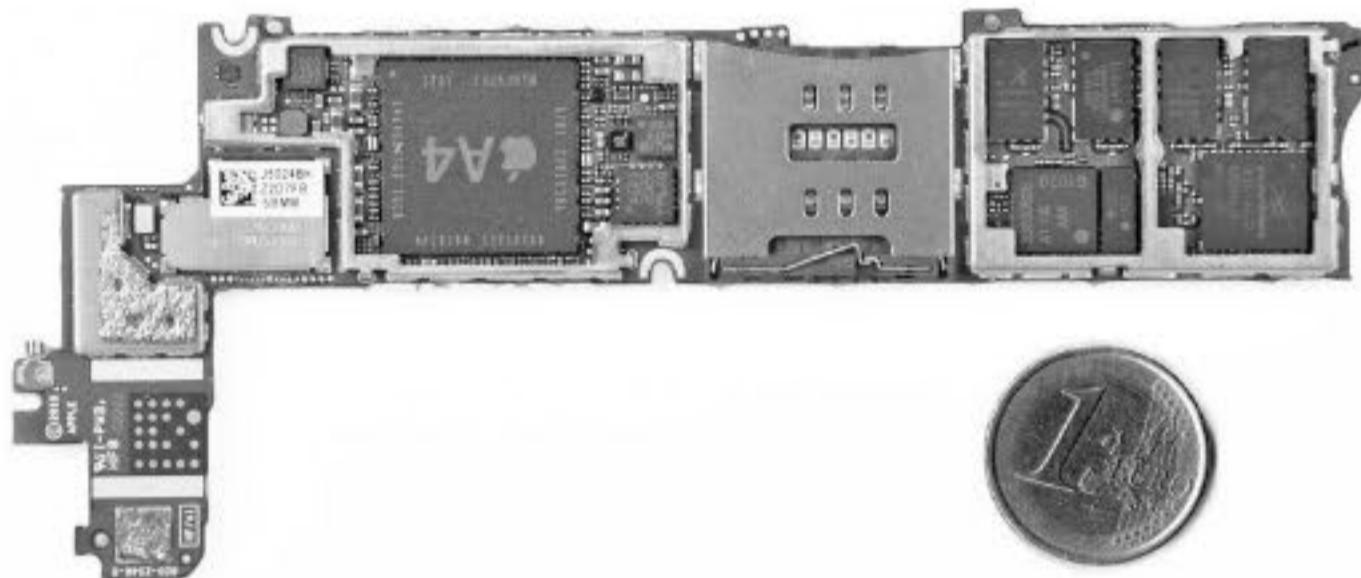


Figura 7.1. Placa principal del Apple iPhone4

Apple sacó a la venta su primer iPhone en el verano de 2007. Desde entonces se han sucedido varias generaciones hasta llegar al iPhone 4, liberado en junio de 2010, y a una versión más avanzada del mismo que bajo la denominación 4s salió al mercado en octubre de 2011. El iPhone 4 funciona con el sistema operativo iOS, versión especial de OSX –instalado en macs portátiles y de sobremesa– adaptada para dispositivos móviles. Su diseño innovador lo distingue

de los modelos anteriores 2, 3G y 3GS. Lleva un lateral metálico que al mismo tiempo actúa como antena. Dispone de una pantalla con 89 mm de diagonal y resolución de 640 x 960 a 326 puntos por pulgada, dos cámaras –una trasera para tomas fotográficas y vídeo y otra frontal para teleconferencia–, giróscopo de 3 ejes, acelerómetro, indicador de contacto con líquidos, flash LED, adaptadores WiFi 802.11 b/g/n, Bluetooth y GPS.

En cuanto a los dispositivos móviles basados en Android, el panorama es bastante más complejo, ya que aunque el sistema operativo es el mismo, existe una variedad enorme de aparatos gobernados por aquél, desde teléfonos de gama baja hasta tabletas con pantallas de alta definición, pasando por ordenadores portátiles, reproductores de medios, equipos Home-TV e incluso relojes de pulsera. La plataforma de hardware principal de Android es la arquitectura ARM. Hay soporte para x86 en el proyecto Android-x86, y Google TV utiliza una versión especial de Android x86. El primer teléfono disponible en el mercado para ejecutar Android fue el HTC Dream, dado a conocer al público el 22 de octubre de 2008. A principios de 2010 Google ha colaborado con HTC para lanzar su producto estrella en dispositivos Android, el Nexus One. Por su parte Samsung, con procesadores ARM y un diseño similar al iPhone, ha puesto en el mercado los dispositivos de la clase Galaxy, que con más de 500.000 aplicaciones disponibles en su tienda de descargas constituye una réplica a los productos de Apple, y probablemente la alternativa que según los expertos se impondrá en número de unidades de aquí al año 2014.

### 7.1.3 Software

Por lo general el sistema operativo que viene instalado en un *smartphone* incluye lo necesario para utilizar el dispositivo con fines particulares e incluso para trabajar en entornos de productividad. Las aplicaciones adicionales se instalan mediante el *App Store* de Apple o las tiendas Android, en caso de que el usuario tenga un aparato de Samsung o HTC. Existen aplicaciones de pago y gratuitas. En el caso de Apple, los programas se descargan directamente al iPhone o a través del software iTunes, utilizado para sincronizar el dispositivo con el ordenador de sobremesa y un número de funciones útiles como actualizar el *firmware*, realizar copias de seguridad, convertir formatos de audio y vídeo, etc. Lo que Apple utiliza en sus dispositivos móviles es una versión adaptada de su sistema operativo OSX para ordenadores de sobremesa y portátiles, con un esquema de particionamiento similar para los soportes de datos e incluso el mismo sistema de archivos HFS+.

Aunque Apple da a conocer las API del sistema y distribuye un kit de programación para desarrolladores –que también utilizará el investigador forense

para acceder a los datos del aparato–, jamás ha querido saber nada sobre instalación de software de terceros. Si un usuario quiere utilizar programas al margen de la *App Store* se verá obligado a “liberar” su iPhone (*jailbreaking*) por medio de utilidades de *hacking* como Ultrasn0w o Pwnage. Estas operaciones implican riesgos de inutilización transitoria (*bricking* o “enladrillado”) del terminal, pérdidas de rendimiento y anulación de la garantía. Por no hablar de una mayor exposición a incidentes de seguridad.

No solo los *hackers* recurren al *jailbreaking*: también la policía se sirve de él para extraer datos de dispositivos incautados en el transcurso de sus investigaciones. Apple se muestra tan reservada en lo referente a la tecnología del iPhone que ni siquiera pone a disposición de la autoridad, ni tiene previsto hacerlo en el futuro, un método adecuado para la recuperación de archivos borrados con fines forenses. Apple no hace esto por codicia ni por desconfianza, sino simplemente para mantener la seguridad en sus productos y redes de datos.

Los dispositivos basados en Android no necesitan *jailbreaking*, porque el concepto de programación en que se basan es diferente. Android es un sistema de código libre basado en Linux que fue desarrollado para su empleo en dispositivos móviles. En 2005 su creadora, la empresa Android Inc., fue comprada por Google y en estos momentos quien mantiene toda la operativa de mantenimiento y actualizaciones del sistema es la Open Handset Alliance, una asociación de desarrolladores de hardware, software y operadores de servicios a la cual pertenecen la misma Google, HTC, Samsung, Dell, Intel, Motorola, Texas Instruments y otras firmas de prestigio. Al igual que en el caso de Linux existe una numerosa comunidad de desarrolladores que se dedican a escribir aplicaciones para mejorar el sistema y ampliar la funcionalidad de los dispositivos. La plataforma más importante para conseguir software es Android Market, una tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de adquirir software de terceros. Los programas están escritos en el lenguaje de programación Java.

Un sistema operativo Android típico se compone de aplicaciones que se ejecutan en un *framework* Java de aplicaciones orientadas a objetos dentro de una máquina virtual Dalvik. La compilación del software se lleva a cabo en tiempo de ejecución, con librerías escritas en C. Los componentes estándar incluyen administrador de interfaz gráfica, *framework* OpenCore, base de datos relacional SQLite, API gráfica OpenGL, motor de renderizado WebKit, motor gráfico SGL, SSL para la conexión segura a servidores web y librerías de C Bionic. Todo ello incluido en alrededor de 13 millones de líneas de código. El motor lo constituye un *kernel* de Linux que se encarga de gobernar funciones básicas de gestión de procesos, administrador de memoria y asignación de servicios a controladores.

Android no se ve libre de problemas de seguridad derivados de su exposición al software malicioso, aunque hasta el momento los únicos incidentes serios han sido provocados por aplicaciones de terceros y no por las de los sitios oficiales, en los que se lleva a cabo una rigurosa política de control de calidad y selección.

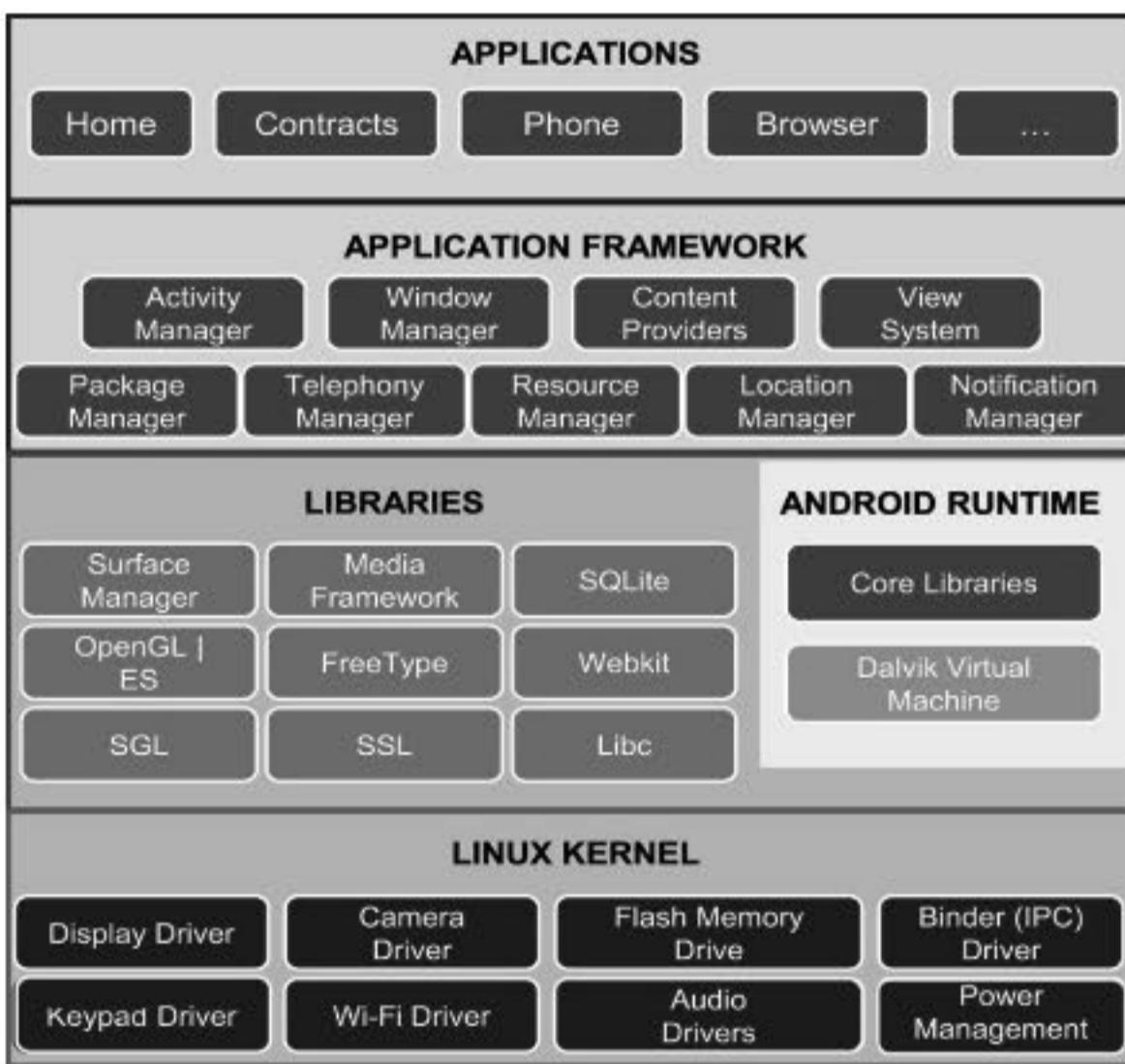


Figura 7.2. Estructura de Android

#### 7.1.4 Información obtenible

Un iPhone o un terminal Android no son como los antiguos teléfonos móviles de los cuales el investigador forense podía rescatar datos de la SIM, una lista de contactos, las últimas llamadas emitidas y recibidas, los textos de algunos mensajes SMS y poco más. Posteriormente el teléfono móvil amplió sus capacidades para incluir archivos multimedia y recursos limitados de Internet y correo electrónico. Nada de esto es sin embargo comparable a la potencialidad de un *smartphone*. Además de la necesidad del investigador en cuanto a ponerse al día en todo lo referente a estos aparatos, conviene que el lector se haga una idea de todo lo que su iPhone o Samsung Galaxy esconde, sobre todo si es ejecutivo de una gran empresa, líder de un partido político o activista comprometido con movimientos cívicos en países con gobiernos dictatoriales.

La investigación de dispositivos móviles es un arte complicado, en el que es necesario improvisar y no hay un conjunto de métodos establecidos. Existen técnicas ingeniosas para puentear contraseñas y otras protecciones. Se puede bloquear la recepción telefónica introduciendo el aparato en una jaula de Faraday (receptáculo con mallas de metal que impide el paso de las ondas electromagnéticas) o simplemente –según dicen– en una bolsa de patatas fritas que tenga el interior forrado con papel de plata. De esta manera se impide el borrado remoto a través de MobileMe y otros servicios de seguridad remota. Aún más fácil: basta poner el iPhone en modo avión (Ajustes), con lo cual el teléfono y la conectividad WiFi quedan inmediatamente desactivados. Algunos dispositivos Android, como por ejemplo los pertenecientes a la gama Galaxy del fabricante surcoreano Samsung, también disponen de una funcionalidad para ponerlos fuera de línea desactivando el teléfono y las conexiones inalámbricas.

El examen de las propiedades básicas –número de serie, IMEI del teléfono móvil, compañía telefónica, etc.–, así como de contenidos básicos como lista de contactos, agenda, fotografías, documentos y demás, resulta trivial. Una vez sorteado mediante diversas técnicas el bloqueo de pantalla del aparato, si lo hubiera, se pueden extraer a un ordenador todos los datos producidos como resultado de la interacción del usuario con su terminal. Más adelante veremos los procedimientos que se emplean para ello, tanto con el iPhone como con dispositivos Android. Mediante un cliente de bases de datos (por ejemplo SQLite, Database Browser o Frog) se puede explorar todo el entramado de tablas en las que un *smartphone* almacena la información. He aquí una muestra de todo lo que el investigador podría hallar en un Apple iPhone:

- Libros de direcciones (*AddressBook*): base de datos principal y más voluminosa del sistema iOS, con informaciones de contacto e imágenes, incluyendo grupos y miembros, números de teléfono, direcciones postales y de correo electrónico.
- Historial de las últimas 100 llamadas telefónicas entrantes y salientes, incluyendo fecha, hora y duración.
- Caché del navegador Safari: información referente a aplicaciones, servicios y fechas de uso de los programas. Dentro de las bases de datos gestionadas por la caché se incluyen datos de antenas de telefonía móvil y puntos de acceso WiFi con los que el dispositivo se va encontrando a medida que el usuario lo traslada consigo de un lado a otro, lo que resulta de gran ayuda para reconstruir itinerarios. Estos datos son imprescindibles para el funcionamiento de *Maps* y otras herramientas de geolocalización.

- Perfiles de configuración creados por el usuario.
- *Cookies*: efectivamente, se trata de esos pequeños archivos de texto almacenados por los navegadores que delatan la actividad del usuario en la red. Como puede ver el usuario no resulta fácil librarse de ellos, ni siquiera en el micromundo del iPhone.
- *Logs* y archivos de registro: información referente al uso de aplicaciones del iPhone.
- Mapas (*Maps*): esta aplicación del iPhone se sirve del motor de Google Maps para localizar y marcar lugares. Combinado con informaciones adicionales –por ejemplo los metadatos Exif de las imágenes tomadas por la cámara fotográfica–, permite conocer la posición geográfica y los trayectos seguidos por el usuario. Requiere tener activa la localización.
- Historial del teclado: guardado en el archivo *dynamic-text.dat*, que funciona como *keylogger* del iPhone y se abre con un simple editor de texto. Información sensible a más no poder. A partir de ella se pueden reconstruir mensajes SMS e incluso correos electrónicos confidenciales almacenados en un servidor remoto.
- Notas: texto escrito a través de *Notes App*, con fechas de creación y modificación de los archivos correspondientes.
- Preferencias: existen informaciones que tomadas por separado dicen bien poco, pero tomadas en conjunto pueden llegar a transmitir un retrato fiel de la personalidad, nivel de ingresos, ideas políticas, modo de vida e incluso estado de salud del usuario. Ajustes y cuentas del correo electrónico, código de país, *App Store*, códigos ICCID, IMSI y ajustes de itinerancia para el teléfono móvil. Aparte de eso el GPS añade información relativa al norte magnético o real.
- También queda constancia de los números para *forwarding* de llamadas, búsquedas recientes en Internet, ajustes de zonas horarias, estado de redes WiFi y Bluetooth, lista de aplicaciones estándar e instaladas por el usuario, cotizaciones favoritas de acciones, lista de ciudades para las cuales se han hecho consultas meteorológicas, vídeos buscados en YouTube y parámetros de todas las redes a las que el iPhone ha estado conectado incluyendo identificadores SSID así como fecha y hora de la última conexión WiFi.

- SMS y MMS: base de datos con mensajes de texto y multimedia enviados y recibidos por el usuario. No solo se guarda el contenido de los mensajes sino también los números de teléfono, fechas, horas y borradores sin mandar.
- Historial de Internet y *bookmarks* del navegador Safari: direcciones de páginas visitadas junto con fecha y hora de acceso. Especial interés posee la funcionalidad de Estado Suspendido, que permite pasar rápidamente de unas páginas a otras hasta un total de ocho que permanecen guardadas en el caché de Safari.
- *Voice-mails* (iPhone OS 3.0 y superior): los mensajes de voz se guardan en archivos con extensión .amr que pueden escucharse con QuickTime.
- *WebClips* y *WebKits*: las aplicaciones de Internet suelen volcar gran cantidad de información a las bases de datos SQLite (recursos URL, nombres de aplicaciones, direcciones de correo electrónico, fechas, etc.).
- Configuración del sistema: en este apartado se incluyen las preferencias del sistema y de la red, junto con las direcciones IP y *hotspots* que el dispositivo va encontrando por el camino.
- Documentos: archivos PDF, Word y de otros formatos: si han sido cargados desde ordenadores portátiles y de sobremesa incluirán metadatos con información referente a otros usuarios, máquinas, impresoras, grupos de trabajo, modificaciones realizadas en el documento junto con los autores y los nombres de las máquinas por las que el documento ha pasado, tiempos trabajados, objetos incrustados como hojas de cálculo Excel, configuraciones de redes, etc.
- Medios audiovisuales: imágenes tomadas con la cámara fotográfica, archivos de sonido MP3, vídeos, grabaciones y recordatorios de voz. En lo que respecta a las fotografías conviene mencionar la información contenida en los metadatos Exif, típicos del formato JPEG y mencionados en un capítulo anterior de este libro, los cuales indican, entre otras cosas, la fecha y hora en que fue tomada la fotografía, modelo de cámara, ajustes, si se disparó el *flash*, software de retoque utilizado, etc. Con las opciones de localización del *smartphone* activadas los metadatos Exif incluyen las coordenadas GPS: latitud, longitud, altura y dirección de la brújula.

- Finalmente todo un caudal de datos relacionados con sitios de la web 2.0 y redes sociales –Facebook, LinkedIn, MySpace, Twitter, etc.–. Así mismo artefactos con información relevante para la seguridad: contraseñas, cuentas bancarias, números de tarjeta de crédito, de la Seguridad Social o de documentos de identidad, registros VoIP, direcciones de correo electrónico, cualificación profesional, cargos y funciones, posición en el organigrama, etc.
- Aunque el ejemplo anterior está basado en un dispositivo de Apple, los terminales Android poseen un potencial comparable para la explotación de elementos de evidencia. Obsérvese que esto no es más que el resultado de una exploración lógica de los recursos y funcionalidades del dispositivo, es decir, lo que el investigador puede obtener de un *backup* convencional del aparato realizado a través del software de sincronización iTunes o el Samsung Kies. Mediante técnicas especiales se pueden obtener permisos de usuario o realizar imágenes a bajo nivel de los dispositivos que permitan excavar todavía más hondo y llegar a una información menos accesible incluyendo los archivos borrados por el usuario. De todo esto se va a tratar en los apartados siguientes.

## 7.2 INVESTIGACIÓN FORENSE DEL APPLE IPHONE

### 7.2.1 Consideraciones generales

Si hace algunos años la investigación de un teléfono móvil de tipo convencional planteaba problemas de diversa índole al investigador –adviértase que estamos hablando de problemas técnicos, porque las cuestiones relacionadas con la valoración jurídica de la evidencia y la admisibilidad de métodos ante los tribunales son un mundo aparte del cual tendremos ocasión de hablar–, la adquisición, análisis forense y realización de informes sobre un *smartphone* supone una tarea enormemente problemática que es necesario abordar por niveles, de manera similar al estudio de las redes informáticas.

Un primer nivel lo constituye la extracción manual, o visualización directa de los datos de un dispositivo incautado tal y como la llevaría a cabo su propietario. Esta operación se documenta mediante tomas fotográficas de la pantalla, realizadas sobre la marcha con una cámara digital, y permitiría acceder a informaciones básicas e inmediatas. En una segunda etapa, después de haber tomado las precauciones adecuadas –como poner el dispositivo en modo vuelo o aislarlo en una jaula de Faraday para evitar un borrado remoto, o bien desactivar la función de bloqueo de la pantalla en caso de que en el preciso momento de la detención el sospechoso lo estuviese manejando– se procedería a una adquisición lógica del

terminal con un ordenador de sobremesa mediante un cable y el software de sincronización o aplicaciones de transferencia de archivos desarrolladas por terceros.

También existe la posibilidad de recuperar información no directamente desde el terminal, sino a partir del *backup* creado por iTunes en el ordenador de sobremesa del usuario. Esto permite adquirir elementos de evidencia no solo en ausencia del dispositivo móvil, sino que en determinados casos se podría recuperar también la versión anterior de un archivo modificado en el terminal. En el nivel correspondiente a las operaciones de adquisición lógica es donde el investigador, mediante el análisis de las bases de datos SQLite, recuperaría las informaciones del listado que figura en el capítulo 7.1.4.

Mucho más exigente y sofisticado es el tercer nivel, al que podríamos denominar de extracción física, y que consiste en la realización de una imagen a bajo nivel del dispositivo, de modo similar a como se llevaban a cabo las adquisiciones forenses de discos duros y soportes de datos explicadas en los capítulos anteriores sobre investigación forense de sistemas MS-Windows y Linux. La extracción física permitiría recuperar archivos borrados con la ayuda de *suites* de software forense como EnCase o utilidades de *data carving* como Foremost o Testdisk. Sin embargo los procedimientos para obtener imágenes a bajo nivel de dispositivos móviles distan de ser triviales y tampoco se hallan libres de complicaciones jurídicas. En el caso del iPhone ya se ha dicho antes que no existe un procedimiento documentado para la adquisición del dispositivo. Hay varios métodos desarrollados por investigadores especializados en el análisis de teléfonos inteligentes, y también otros que implican la realización previa de maniobras de *jailbreaking* y por consiguiente alteraciones en la evidencia, lo cual siempre resulta problemático ante un tribunal.

En niveles superiores de extracción de datos avanzaríamos hacia el empleo de técnicas que solo están al alcance de técnicos especializados en microelectrónica y un laboratorio bien equipado, como por ejemplo el volcado directo del contenido de los chips de memoria, tras haberlos extraído de sus placas, o la microlectura o interpretación directa de las puertas físicas del circuito. Estos procesos son costosos en recursos económicos y tiempo, y requieren un conocimiento profundo de aspectos técnicos relacionados con la memoria *flash* y los sistemas de archivos. Tampoco existen herramientas comerciales para llevarlos a cabo. Su empleo solo se justifica en situaciones excepcionales o en caso de que la investigación tenga que ver con un asunto de extrema importancia.

Lo que el lector va a encontrar aquí no es más que una descripción muy general y resumida de unas operaciones que en el fondo son bastante más complicadas de lo que pudiera imaginar. Existen distintas variantes del iPhone y un

buen número de versiones del software iOS. La tabla de referencias cruzadas entre unas y otras, con las diferentes posibilidades, particularidades técnicas y limitaciones dependiendo de la combinación hardware+software es demasiado extensa como para incluirla en una obra introductoria como esta. Si desea hacer un buen trabajo en la investigación forense de dispositivos móviles, lo que tiene que hacer es en primer lugar leer alguno de los libros especializados en este campo que figuran en la bibliografía, luego asistir a algún cursillo sobre la materia impartido por empresas especializadas y finalmente adquirir experiencia realizando pruebas con terminales de segunda mano. No intente ensayar lo que aquí se cuenta con un dispositivo móvil perteneciente al inventario de pruebas porque los resultados podrían ser desastrosos, con unas repercusiones jurídicas demoledoras para su posición en el caso.

## 7.2.2 Adquisición del iPhone mediante iTunes

En el nivel más elemental se puede obtener una cantidad apreciable de datos con un proceso de sincronización simple del iPhone a través de iTunes. Este método funciona más o menos con todas las versiones del dispositivo y de iOS. Puede haber dificultades en caso de que el aparato esté protegido por un código de acceso o se utilice encriptación de datos para hacer el *backup*. Existen diversas técnicas para sortear el primer inconveniente. En cuanto al segundo, la encriptación de particiones, esta funcionalidad solo está disponible a partir del sistema operativo iOS 4.0. Cualquier aparato de la clase 3 o 3G podrá ser adquirido de modo simple y sin grandes problemas mediante este procedimiento. Del mismo modo iTunes puede utilizarse para la investigación de otros dispositivos Apple como los iPod, iTouch o las tabletas iPad.



Figura 7.3. Apple iTunes

Previamente a la sincronización de un iPhone mediante iTunes el investigador deberá familiarizarse con este software para no cometer un error que no solo podría frustrar la operación de respaldo del terminal, sino borrar los datos que este contiene, perdiéndose de este modo elementos de evidencia valiosos para el caso. Una sincronización irreflexiva o mal planificada puede traer consigo, en lugar del deseado trasvase de los archivos del dispositivo al ordenador de sobremesa, una destrucción accidental de los mismos si no se tienen en cuenta las opciones de prioridad, el estado de las aplicaciones y otros parámetros de ajuste.

Antes de tomarse este aviso a la ligera le recomendaría que se lea algunas de las innumerables quejas planteadas en foros de Internet por usuarios de iPhone, iTouch o iPad que aseguran haber perdido sus canciones y sus libros electrónicos misteriosamente tras un proceso de sincronización o actualización del *firmware*. Aunque dé la impresión de que algún misterioso duende o la ley de Murphy les están jugando una mala pasada lo más probable es que no leyeron con atención todos esos mensajes que iTunes muestra en pantalla advirtiendo al usuario que como resultado de la sincronización podrían eliminarse datos en el terminal, o si quiere que haga una copia de seguridad de las aplicaciones compradas en el *App Store* y otras preguntas por el estilo.

Una vez instalado iTunes (en un ordenador Apple OSX o un PC con cualquier versión superior a Windows XP inclusive) el software se iniciará automáticamente. Probablemente exista una versión más avanzada de *firmware* para el terminal. iTunes preguntará al usuario si quiere descargarla y actualizarla. En cualquier circunstancia el investigador deberá responder que no: la actualización lleva su tiempo, ya que implica descargar una partición entera del sistema operativo iOS y escribirla sobre la ya existente en el iPhone. Aparte de eso se perderán las aplicaciones y los datos, o como mínimo podrían quedar afectados por algún cambio. La sincronización suele ser automática. En este punto es necesario que el investigador ponga toda su atención en lo que hace. Normalmente la primera vez que conecta el dispositivo iTunes desmarca por defecto todas las opciones de sincronización: música, videos, libros, etc.

También le advertirá que hay en el terminal elementos comprados en *App Store*, aconsejándolos que haga una copia de los mismos en el disco duro. La primera vez que lo intente fracasará al no disponer de permisos de acceso en el ordenador. Para adquirirlos será necesario disponer de los datos correspondientes a la identidad Apple del sospechoso (dirección de correo electrónico y *password*). Si no los ha podido conseguir no importa, porque mientras no se haya especificado una sincronización de aplicaciones, iTunes deja intactas las copias de respaldo que hay en el dispositivo y realiza un nuevo *backup*.

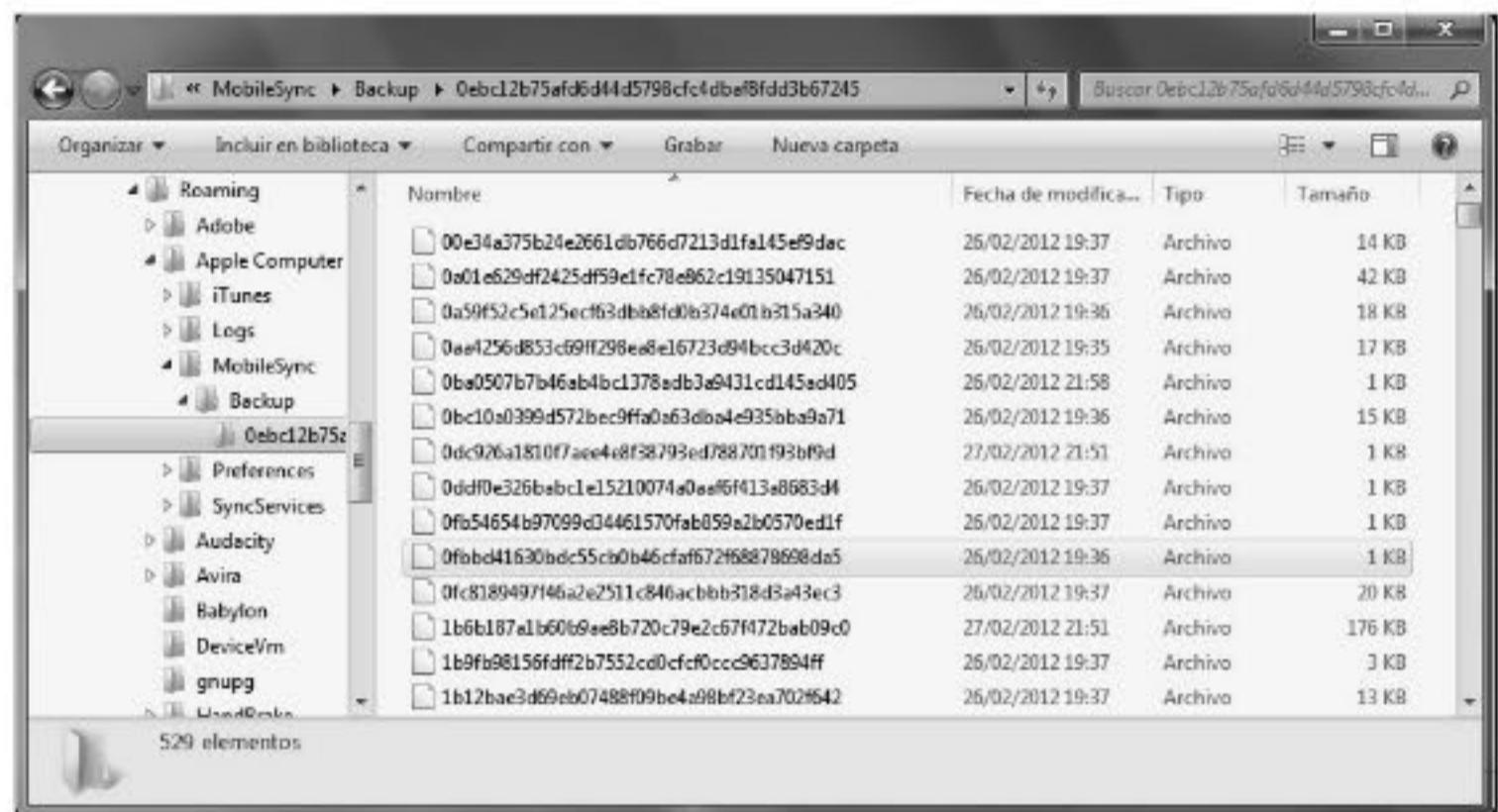


Figura 7.4. Aspecto poco legible de un backup de iTunes

### 7.2.3 iPhone Backup Extractor

Una vez llevada a cabo la sincronización y adquirida la copia de respaldo, al investigador se le presentan dos opciones. Una de ellas consiste en analizarla directamente en su emplazamiento del disco duro, lo cual puede resultar bastante complicado porque iTunes guarda sus datos en un formato que está pensado para su propio uso interno y no para hacerle la vida cómoda a los investigadores forenses. La ubicación por defecto de la copia de respaldo varía en función del sistema operativo. Estas son las más habituales:

```
Macintosh HD > Users > Usuario > Library > Application Support > MobileSync > Backup

Windows XP:
\Documents and Settings\username\Application Data\Apple Computer
\MobileSync\Backup\

Windows Vista/7:
\Users\username\AppData\Roaming\Apple Computer\MobileSync\Backup\
```

Como se puede ver en la figura 7.4, la copia de seguridad consta de una carpeta con numerosos archivos cuyos nombres están compuestos por secuencias de caracteres hexadecimales aparentemente aleatorios y nada fáciles de recordar. Para poder hacer con ellos algo útil el investigador puede recurrir a los servicios de una aplicación de terceros llamada *iPhone Backup Extractor*, y que es capaz de abrirse paso entre ese laberinto de datos convirtiendo los archivos a formatos manejables mediante aplicaciones estándar como hojas de cálculo, visores de

gráficos, reproductores de vídeo o herramientas de gestión de bases de datos. *iPhone Backup Extractor* se puede conseguir en la dirección <http://www.iphonebackupextractor.com/> y dispone de versiones para Windows, OSX y Linux. Una vez instalado su manejo es sencillo. Inmediatamente después de arrancar buscará automáticamente las ubicaciones por defecto de las copias de respaldo. En caso de que iTunes hubiera almacenado los *backups* en otro lugar habrá que indicárselo al software.

*iPhone Backup Extractor* recupera los datos del terminal (contactos, listas de llamadas, *cookies*, coordenadas geográficas) en forma de tablas de bases de datos que se pueden visualizar sin problemas con cualquier cliente SQLite. También permite recuperar imágenes y archivos de vídeo. Una ventaja adicional de *iPhone Backup Extractor* reside en el hecho de que para recuperar los datos del dispositivo móvil no es necesario que este se encuentre conectado al ordenador. A veces ni siquiera es necesario que aparezca en el registro. Basta con apoderarse del ordenador de sobremesa o portátil del sospechoso, y si este ha realizado en algún momento una sincronización de su terminal, existen muchas probabilidades de que la copia de respaldo correspondiente se encuentre allí guardada a la espera de que el investigador e *iPhone Backup Extractor* vengan a dar con él. Se trata de un software de pago, pero su versión de evaluación resulta lo bastante eficaz como para que el investigador se pueda hacer una idea de sus capacidades y ventajas.



Figura 7.5. iPhone Backup Extractor

## 7.2.4 Acceso a un backup encriptado

El usuario podría haber iniciado un *backup* encriptado mediante la opción de poner una contraseña antes del proceso de sincronización. En este caso los datos serán de poca utilidad para el investigador a menos que este utilice una herramienta para *cracking* de contraseñas, como por ejemplo *iPhone Password Breaker* de Elcomsoft. Este software proporciona accesibilidad forense a las copias de respaldo de terminales iPhone e iPad protegidas por contraseña. Para ello recupera el *password* en texto claro utilizado para encriptar los archivos del *backup*. A partir de la versión 4 de iOS la encriptación de *backups* se lleva a cabo mediante hardware, pero *iPhone Password Breaker* aún es capaz de leer y desencriptar el *keychain* –un área reservada donde se guardan contraseñas y otras informaciones sensibles– del iPhone. Este software soporta todas las versiones de iOS, con lo cual la característica de encriptación avanzada no debería ser un obstáculo a la hora de obtener la contraseña.

Para usar *iPhone Password Breaker* no es necesario tener el terminal a mano, sino únicamente los archivos encriptados de la copia de respaldo (particularmente un archivo llamado “Manifest.plist”). Al ejecutar la herramienta aparecerá la pantalla principal que se puede ver en la figura 7.6. A partir de ella (**Open**) el investigador podrá ver la lista de dispositivos respaldados en el ordenador. Los que le interesan son aquellos que están encriptados, estado que se indica mediante el símbolo del candado. La contraseña se obtiene mediante ataques de fuerza bruta, pudiendo elegirse un número determinado de caracteres alfanuméricos para minimizar el tiempo de búsqueda.

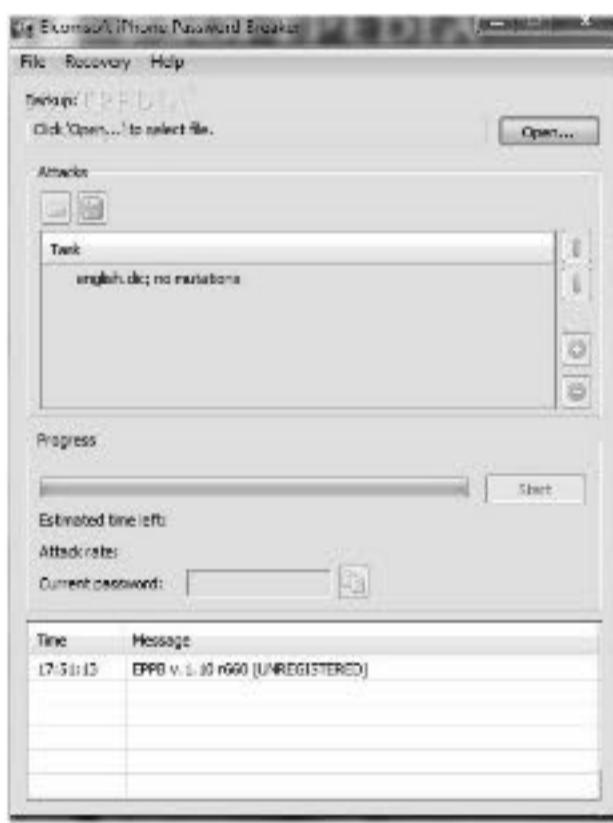


Figura 7.6. *iPhone Password Breaker*

## 7.2.5 Adquisición lógica con herramientas de terceros

La adquisición lógica de un dispositivo, como ya se ha comentado en más de una ocasión, consiste en una lectura sistemática de todos los archivos que el mismo contiene y a los que el sistema asigna de manera explícita bloques de datos en el volumen de almacenamiento. Esto es válido no solo para terminales móviles, sino en general para todo tipo de soportes de almacenamiento. En la investigación forense del iPhone –así como en la de aparatos provistos de Android y otros sistemas móviles– los procedimientos de adquisición lógica constituyen la primera alternativa por un número de razones. En primer lugar son fáciles de llevar a cabo. Para ello, además, se dispone de herramientas de terceros que además de extraer los archivos pueden convertirlos a formatos de aplicaciones estándar e incluso crear informes forenses de manera automática. Con frecuencia una adquisición lógica así mismo permitirá al investigador forense los elementos de evidencia suficientes para poder ayudar a sostener un caso delante de los tribunales. Cuando no sea así habrá que recurrir a técnicas de extracción de datos más sofisticadas.

El método de trabajo en una adquisición lógica, independientemente del software empleado, viene a ser siempre más o menos el mismo, y consta de cuatro etapas consecutivas:

1. Ejecución de la herramienta seleccionada por el investigador.
2. Conexión del terminal a través de un cable.
3. Realización de la copia lógica: mediante la misma se extraen del dispositivo los mismos datos que en una operación de respaldo llevada a cabo por iTunes, con la única diferencia de que en este caso el investigador accede directamente al dispositivo.
4. Análisis de la información recuperada y elaboración del informe.

Existen numerosas herramientas para la extracción lógica de archivos y datos del iPhone. En los departamentos de delitos informáticos de la policía, laboratorios forenses que trabajan para la administración de justicia y departamentos de seguridad de las grandes empresas se suelen utilizar soluciones integradas de carácter comercial que permiten adquirir dispositivos móviles Apple y de otras marcas, tanto modernos como antiguos: desde las ya obsoletas agendas electrónicas Palm Pilot hasta teléfonos móviles de gran número de modelos y operadores celulares de todo el mundo. Tal es el caso de algunos paquetes de software como Oxygen Forensics Suite 2010, XRY, Lantern, Paraben Device Seizure o EnCase Neutrino, o dispositivos hardware especiales como Cellebrite UFED (*Universal Forensics Extraction Device*).

Habrá ocasiones en las que una extracción lógica de archivos no resulta suficiente para alcanzar los objetivos de una investigación. Tal es el caso cuando interesa obtener informaciones relativas al propietario anterior del dispositivo, o si se sospecha que este ha podido borrar archivos de su dispositivo móvil. Entonces, y dependiendo de las circunstancias, puede ser necesario recurrir a una adquisición física del dispositivo. Ello implica la realización de una imagen a bajo nivel, con el espacio no asignado y aquellos archivos procedentes de una actividad anterior del usuario que todavía no hayan sido sobreescritos con datos nuevos. Aquí es donde aparecen las dificultades y se comienza a tener que hacer frente a los más diversos riesgos, tanto para la integridad de los datos con los que se trabaja como delante del tribunal, porque a diferencia de los soportes de datos para ordenadores de sobremesa, examinados en capítulos anteriores, no existen procedimientos normalizados ni de amplia aceptación para la realización de imágenes forenses a partir de dispositivos móviles.

La copia a bajo nivel de un disco duro, una llave USB o un DVD-ROM se puede llevar a cabo sin problemas y con garantías plenas de no alterar en lo más mínimo el medio de origen. No sucede lo mismo con las memorias de estado sólido NAND donde los teléfonos inteligentes y dispositivos móviles guardan sus datos. Para extraer su contenido con frecuencia es preciso aplicar técnicas intrusivas que implican algún tipo de alteración en el soporte, como se podrá apreciar en el siguiente ejemplo de adquisición física.

## 7.2.6 Adquisición física de un iPhone

Recordemos que adquirir físicamente un dispositivo significa realizar una imagen a bajo nivel del mismo, trasvasando a un soporte de destino no solo los archivos que contiene y que podríamos haber recuperado mediante alguno de los procedimientos lógicos descritos con anterioridad, sino también los archivos borrados y los datos existentes en sectores no asignados del medio así como en aquellas partes del mismo que no resultan accesibles a través de las estructuras del sistema de archivos. Seguramente el lector se estará preguntando por qué esto no es tan fácil con un iPhone como en el caso de los discos duros o las llaves USB, según se pudo ver en capítulos anteriores: conectar el dispositivo a un puerto USB y crear una imagen en *bitstream* con dd o Adepto.

El principal inconveniente está en que el terminal no es un receptáculo pasivo de datos, sino un ordenador en miniatura con su propio sistema operativo corriendo sobre un conjunto de chips integrados RAM/NAND que necesariamente tiene que estar funcionando para llevar a cabo cualquier operación, incluso para

extraer datos del dispositivo. Desgraciadamente los mecanismos de seguridad del iPhone impiden realizar una adquisición a bajo nivel sin obtener privilegios de acceso, lo cual resulta imposible con un terminal recién salido de fábrica con su equipamiento de software y herramientas de productividad estándar.

Existen varios procedimientos ideados por algunos de los especialistas que se han dedicado a estudiar a fondo la tecnología del iPhone e iOS con finalidades forenses. La operativa es bastante compleja y no exenta de riesgos. Uno de ellos ha sido desarrollado por Jonathan Zdziarski, antiguo investigador de la empresa de seguridad informática y antivirus McAfee, Inc. y autor de varios libros sobre tecnología e investigación forense del iPhone. Este procedimiento requiere la descarga previa desde la página web de Zdziarski de herramientas que permiten transferir al iPhone un agente de software capaz de iniciar con el ordenador del investigador forense un canal de comunicación a través del cual transferir la imagen del dispositivo. Como resultado de todas estas operaciones se crea un archivo de imagen con extensión dmg que el investigador podrá analizar con TSK u otras herramientas forenses en busca de archivos borrados, cadenas de caracteres y otros elementos de evidencia.

## 7.2.7 Jailbreaking

Más al alcance del usuario –pero solo recomendable como ejercicio y a condición de no emplear ningún dispositivo que haya sido catalogado como prueba forense– se halla la adquisición física del iPhone mediante el empleo de técnicas de liberación o *jailbreaking* similares a las que emplean los usuarios avanzados para poder instalar en sus aparatos software de terceros al margen de la *App Store* de Apple. En caso de que el investigador se anime a llevar a cabo la prueba, se recomienda que lo haga con un terminal de su propiedad y que no le sea imprescindible para sus necesidades habituales, ya que cualquier cambio en la partición del sistema operativo implica automáticamente una anulación de la garantía de Apple. Hasta hace poco, incluso, el *jailbreaking* estaba considerado como una práctica delictiva por presiones de Apple, hasta que hace un par de años los tribunales lo despenalizaron mediante una resolución que fallaba en contra de esa cláusula abusiva de la empresa de Cupertino. En la actualidad el usuario puede liberar su terminal sin temor a otras represalias aparte de la anulación de la garantía. Los métodos para ello se hallan bien documentados y están disponibles en Internet.



Figura 7.7. Aplicación típica de Cydia: un terminal de texto para el iPhone

El *jailbreaking* deja el dispositivo libre para instalar en él aplicaciones de terceros desde tiendas como Cydia. Con ello el usuario logra extender la funcionalidad de su iPhone más allá de los ámbitos autorizados por Apple, como por ejemplo la instalación de consolas de texto que permiten introducir comandos para interactuar con el sistema operativo al más puro estilo Unix (figura 7.7).

## 7.2.8 Adquisición basada en técnicas de jailbreaking

El *jailbreaking* produce cambios en la partición del sistema operativo del iPhone, vulnerando así un principio maestro en la práctica forense: la no alteración de las pruebas. También existe la posibilidad de que algunos elementos de evidencia queden sobreescritos, con lo cual cabe pensar que la adquisición de un dispositivo móvil mediante esta técnica solo sería autorizada por un tribunal en circunstancias excepcionales y con el requisito estricto de que el investigador dejara una constancia documental minuciosa y exacta de todos los pasos seguidos en la operación. Por supuesto en estas páginas no vamos a dar indicaciones para liberar un iPhone. Hay muchas páginas de Internet en las que se habla detalladamente sobre temas de *jailbreaking* y se dan las instrucciones precisas para llevar a cabo la maniobra con seguridad, una vez determinado el modelo de dispositivo y la versión del software. Preferimos partir del supuesto de que el terminal ya está liberado y el usuario puede hacer con él lo que seguidamente se explica, que a grandes rasgos consta de tres fases: en primer lugar establecimiento de una red inalámbrica; segundo, conectar al iPhone desde un ordenador de sobremesa o portátil Apple con el sistema operativo OSX; y finalmente realizar la imagen a bajo nivel del dispositivo móvil.

**Crear una red inalámbrica:** el objetivo consiste en lograr una conexión remota con el iPhone. Para ello es necesario establecer una red inalámbrica –a no ser que se disponga ya de una– mediante el menú de preferencias de red de un sistema OSX. En nuestro ejemplo la red que necesitamos llevará el nombre “iPhone-Book”. Lo siguiente consiste en asignar direcciones IP estáticas tanto al ordenador de sobremesa como al iPhone. En un Mac esto se consigue abriendo una ventana de terminal y tecleando lo siguiente:

```
$ sudo ifconfig en1 inet 192.168.1.1 netmask 255.255.255.0
```

No olvide que por debajo de su interfaz gráfico OSX es en realidad una versión adaptada de Unix BSD. Por ello las órdenes que acabamos de introducir en línea de comando son tan parecidas a las que emplearíamos en Linux. Del mismo modo el sistema, antes de ejecutar el comando, le exigirá que introduzca su contraseña de usuario. A continuación hay que establecer otra IP estática en el iPhone. Para ello es necesario ir a **Ajustes → WiFi** y desde allí conectarnos a la recién creada red inalámbrica. Siga la flecha azul y pulse la opción **Estática**, a la derecha de la nueva pantalla que aparecerá deslizándose desde la derecha, para introducir en los apartados correspondientes, por ejemplo, 192.168.1.2, y la misma máscara de subred 255.255.255.0. Una vez completados estos pasos, pulse **Redes WiFi** para volver a la ventana anterior dejando guardados los ajustes de conexión. Una vez en la misma subred, el ordenador y el iPhone deberían poder comunicarse sin problemas, tras de lo cual ejecutariamos la segunda fase.

**Conexión remota al iPhone:** ahora el investigador puede conectarse remotamente al dispositivo móvil mediante SSH. Dependiendo del procedimiento de *jailbreak* seguido, el nombre del usuario con privilegios de administrador y la contraseña suelen variar. Suponiendo que sean “root” y “alpine”, como suele suceder en la mayor parte de los casos, la conexión al iPhone se establece ejecutando en el Mac los comandos que se van a ver a continuación. Antes de continuar es necesario decir que como parte del proceso de *jailbreaking*, y en particular para el propósito que nos ocupa, conviene que el investigador haya instalado en su dispositivo móvil los paquetes recomendados por el sitio de *jailbreak* para obtener un acceso privilegiado al terminal. Si el investigador no ha completado este paso, conviene que dedique algún tiempo a familiarizarse con el manejo de Cydia y sitios de descargas por el estilo con vistas a la instalación de OpenSSH y otras herramientas que pudieran resultar de interés:

```
$ ssh root@192.168.1.2
root@192.168.1.2's password:
3GS-40: ~root#
```

Una vez conectado el dispositivo hay que cerciorarse de que a bordo de este último se encuentran las herramientas que vamos a necesitar para realizar la imagen y transferirla al Mac. Quizás lo haya adivinado: se trata de dd y netcat. Para saberlo teclee lo siguiente:

```
3GS-40: ~root# which dd
3GS-40: ~root# which nc
3GS-40: ~root#
```

Esto indica la ruta de los dos comandos. Si no muestran ninguna salida eso quiere decir que no se instalaron en el iPhone durante la operación de *jailbreaking*. En cualquier caso, ahora que tiene una conexión SSH podrá trasladarlos desde el Mac. Puesto que tanto este como el dispositivo móvil utilizan el mismo sistema operativo no hay problema para que funcionen tanto en uno como en otro. Para moverlos desde el Mac, en primer lugar tenemos que localizar su ruta:

```
Forensic-Macbook: ~forensic# which dd
/bin/dd
Forensic-Macbook: ~forensic# which nc
/usr/bin/nc
Forensic-Macbook: ~forensic# scp /bin/dd root@192.168.1.2:/bin/dd
Forensic-Macbook: ~forensic# scp /usr/bin/nc root@192.168.1.2:/bin/nc
```

**Adquisición a bajo nivel:** la imagen se realiza mediante dd, y el archivo resultante se crea en el Mac a través de un túnel TCP abierto por netcat a través de un puerto especificado. Para esta operación es preciso establecer en primer lugar el túnel y a continuación ejecutar la copia a bajo nivel. En primer lugar se habilita el extremo receptor en el Mac:

```
Forensic-Macbook: ~forensic# nc -l 9000 | dd of=~/Escritorio/rdisk0s2.dmg bs=1048576
```

Utilizando el operador de concatenación este comando ordena a netcat escuchar en el puerto 9000 y depositar en el escritorio del ordenador un archivo denominado rdisk0s2.dmg con los datos en bruto que van saliendo al otro extremo del túnel. Elegimos la extensión dmg para que, siendo la imagen de la partición de datos de usuario HFS+ de un iPhone, OSX pueda manejarla como una unidad de disco convencional y se pueda examinar su interior con solo montarla. El paso siguiente consiste en ir al dispositivo móvil y listar los archivos representativos de las particiones para saber cuál es el que tenemos que introducir como entrada de dd. No por casualidad hemos llamado antes rdisk0s2 a la imagen que se va a crear en el Mac. iOS emplea un sistema de identificación de particiones típico de Unix. Las unidades de disco –se le llama así aunque se trate de una unidad de memoria NAND– se denominan rdisk0, rdisk1, etc. Y dentro de estas las particiones también poseen, como las del mundo Unix/Linux, sus nombres de archivo característicos: rdisk0s1, rdisk0s2, etc. Veámoslo en el ejemplo de este iPhone:

```
3GS-40: ~root# ls -l /dev/rdisk*
crw-r--- 1 root operator 14, 0 Dec 21 20:43 rdisk0
crw-r--- 1 root operator 14, 0 Dec 21 20:43 rdisk0s1
crw-r--- 1 root operator 14, 0 Dec 21 20:43 rdisk0s2
crw-r--- 1 root operator 14, 0 Dec 21 20:43 rdisk0s2s1
```

La partición rdisk0s2s1 es para uso especial de sistema y característica a título exclusivo del modelo 3GS. La que interesa al investigador es la partición de datos de usuario rdisk0s2, que será la que tenga que utilizar para llevar a cabo la adquisición a bajo nivel del volumen de datos que necesita para su investigación forense. Es el momento de completar el comando que habíamos tecleado antes en el Mac, dejando de este modo abierto en el iPhone el otro extremo del túnel:

```
3GS-40: ~root# /bin/dd if=/dev/rdisk0s2 bs=1M | /bin/nc 192.168.1.1 9000
```

Expliquemos qué sucede: dd copia la partición de usuario –incluyendo el espacio sin asignar, archivos borrados y cualesquiera otras estructuras de datos que pueda haber allí– en bloques de 1 MB y mediante concatenación (operador "|") los transfiere a netcat, el cual a su vez bombea los datos a través del túnel hasta el ordenador de destino (en la dirección IP 182.168.1.1). A bordo de este la instancia de netcat que dejamos escuchando el puerto 9000 los recogerá en la misma cadencia de 1 MB y los irá alineando en un archivo llamado rdisk0s2.dmg. Sabrá que el proceso se está desarrollando con éxito cuando vea crecer el tamaño del archivo en el escritorio.

Dependiendo de si su iPhone es de 8, 16 o 32 GB se verá obligado a esperar más o menos tiempo hasta que la operación esté completa. En el mejor de los casos –¡cuidado con posibles perturbaciones de la transmisión inalámbrica, como por ejemplo una distancia excesiva o demasiado corta entre los dispositivos, paredes de determinados materiales o fuentes de interferencias electromagnéticas!– el investigador tendrá que esperar entre una o dos horas. Una vez finalizado el proceso, en el ordenador Mac tendrá una imagen de la partición de usuario del iPhone, que podrá abrir directamente con las utilidades del sistema, analizar con TSK u otras utilidades forenses. Así mismo la imagen podrá ser trasladada a un PC para procesarla con *suites* forenses como EnCase o FTK o cualquier herramienta de tallado de archivos como Foremost o Scalpel.

## 7.2.9 Adquisición de otros dispositivos Apple

El iPhone no es el único producto de la casa Apple que reviste interés forense por su capacidad para ser utilizado como soporte de almacenamiento de datos o herramienta informática para operaciones de *hacking* e intrusión en sistemas. También son objeto de investigación los diferentes modelos de iPod producidos hasta la fecha –particularmente el iPod Touch con su memoria de

estado sólido de hasta 64 GB y conectividad WiFi integrada—, el dispositivo denominado Apple TV y las tabletas iPad. Estas últimas pueden ser adquiridas mediante procedimientos lógicos a través de iTunes y con herramientas de terceros de manera similar al iPhone. El procedimiento de adquisición física mediante técnicas de *jailbreak* que acabamos de explicar para el iPhone también es aplicable al iPad.

Lo mismo se puede decir en términos generales del iTouch, que ejecuta un sistema operativo iOS y se distingue del iPhone únicamente por la ausencia de conectividad GSM. En principio se le pueden aplicar todas las técnicas descritas con anterioridad así como otras que también forman parte del arsenal de investigación forense de dispositivos móviles. En cuanto a Apple TV, este aparato básicamente consiste en un disco duro que ejecuta una versión modificada de OSX, y puede ser adquirido y analizado por los métodos forenses habituales.

### 7.3 DISPOSITIVOS ANDROID

La investigación forense de ordenadores de sobremesa con Windows y Linux parece una actividad que el investigador puede dominar y llevar a cabo aplicando procedimientos estándar y un catálogo de buenas prácticas. El análisis del iPhone, pese a las complicaciones técnicas y jurídicas, aún se puede abordar bajo las mismas premisas. Sin embargo el panorama cambia por completo cuando se trata de la adquisición y el examen de dispositivos basados en el sistema operativo Android. Una razón para ello, y no precisamente la menos importante, es el ritmo vertiginoso al cual se suceden las versiones de Android: nada menos que 15 niveles consecutivos de API, desde el “primitivo” Android 1.0 liberado el 23 de septiembre de 2008, hasta el más reciente 4.0 (*Ice Cream*) con API 14 – 15 de mayo de 2011, pasando por las versiones 2.2 (*Froyo*) y 2.3 (*Gingerbread*) de mayo y diciembre de 2010 respectivamente, que en la actualidad se encuentran instaladas en el mayor número de dispositivos móviles y teléfonos inteligentes.

A la complicación de todas estas versiones del sistema, con características funcionales y prestaciones distintas, se añade la enorme variedad de marcas comerciales, modelos y tipos de dispositivo equipados con Android, desde teléfonos a reproductores de medios e incluso electrodomésticos. El elevado número de sistemas de archivos reconocidos por el *kernel* Linux y utilizados en la práctica por el entorno Android según para qué funcionalidad también supone una dificultad a tener en cuenta. Aunque Android es solamente uno, basado en Linux y ajustado a estándares estrictos, su carácter multifacético, unido a las complicaciones jurídicas de las que anteriormente se ha hablado relacionadas con la inevitabilidad de alterar el soporte de datos para llevar a cabo determinadas tareas de análisis forense, hace que la investigación de dispositivos móviles

basados en este sistema operativo haya de ser necesariamente cometido de especialistas que tengan práctica en la manipulación de aquellos. Una visión general de este campo y consejos relacionados con la prudencia es por tanto lo más a que pueden aspirar las páginas siguientes.

#### 7.3.1 Introducción a Android

Android es un sistema operativo desarrollado por la Open Handset Alliance (OHA), organización compuesta por más de 50 empresas del sector de las comunicaciones móviles en la cual figuran desde fabricantes de teléfonos y proveedores de servicios (como Samsung, Ericsson, China Mobile y Telefónica) hasta fabricantes de semiconductores (Qualcomm y Texas Instruments) y desarrolladores de software (eBay, NXP). El objetivo de este consorcio, según hace constar en sus propios documentos fundacionales, consiste en “acelerar la innovación en el campo de las comunicaciones móviles ofreciendo al consumidor una experiencia de conectividad más gratificante, económica y de mayor calidad”. El proyecto está respaldado estratégicamente por Google, que patrocina bajo su marca incluso la fabricación de terminales y otros dispositivos móviles equipados con Android.

¿Qué interés puede tener Google en fomentar el desarrollo de una plataforma para dispositivos móviles, a través de un esfuerzo compartido con semejante variedad de consorcios industriales al que anima a unir fuerzas y trabajar de manera coordinada en la creación de un sistema operativo de código libre? Se ha pensado en la posibilidad de que Google quisiera convertirse en un operador de telefonía móvil. Sin embargo la respuesta tiene que ver más con la actividad principal de la compañía que con ambiciosas aventuras empresariales en terrenos ajenos. Google está convencida de que su misión consiste en organizar la información existente en todo el mundo. La disponibilidad de plataformas económicas, seguras y bien diseñadas constituye un requisito indispensable para el logro de los fines de Google, que no son otros que los de ganar dinero gracias a los ingresos de la publicidad y los servicios relacionados con la búsqueda de información.

La arquitectura básica de Android está basada en el núcleo 2.6 de Linux, que gestiona el hardware –teléfono móvil, reproductor MP3 o cualquier otro aparato– a través de una batería de controladores. Sobre esta capa de base encontramos otras con librerías y servicios de software diversos hasta llegar a las aplicaciones con las que el usuario está familiarizado, como el teclado del teléfono, la libreta de contactos, el navegador de Internet o las *apps* que se descarga de su tienda virtual Android. La diferencia con un sistema operativo tradicional, sin embargo, se encuentra en la interposición de un nivel especial compuesto por el

Runtime Android y una máquina virtual en miniatura denominada Dalvik, en la cual se ejecutan aplicaciones originariamente escritas por sus desarrolladores en Java, y convertidas después al formato ejecutable de Dalvik mediante archivos con extensión “.dex”. Cada aplicación corre en su propia máquina virtual y no comparte sus datos con otras. El motivo de este planteamiento de diseño, que implica la introducción de conceptos de *cloud computing* y virtualización en el micromundo de la informática móvil, atiende más que nada a consideraciones fundamentales de seguridad. Esta seguridad se consigue mediante un sistema de permisos en el mismo nivel de ejecución de los procesos, asignando a las aplicaciones identificadores de usuario y grupo. Ninguna aplicación puede interferir con otra a no ser que explícitamente se le concedan permisos para hacerlo.

### 7.3.2 Adquisición de la tarjeta de memoria

Los mecanismos de seguridad de un terminal basado en Android pueden dificultar el acceso lógico y la adquisición física, pero en aquellos dispositivos que disponen de un *slot* para tarjeta de memoria –SD o MicroSD– hay una operación que el investigador forense puede llevar a cabo de manera trivial: realizar una imagen a bajo nivel del soporte de datos. Para ello basta extraer la tarjeta de memoria y pasársela a la estación de trabajo forense por medio de un adaptador USB. En la mayor parte de los terminales Android las tarjetas se utilizan para ampliar la capacidad del dispositivo, por lo que es muy probable que dentro de ella existan datos o aplicaciones descargadas de Android Market e instaladas allí para no saturar el espacio interno del terminal.

La adquisición se lleva a cabo mediante dd o cualquier otro de los métodos estudiados en el capítulo 3. Una vez adquirida la tarjeta de memoria no es conveniente volverla a insertar en el terminal, sino que el investigador debe reemplazarla por otra vacía del mismo tipo para proseguir con las tareas de análisis. La tarjeta con los datos del usuario será precintada y agregada al inventario de pruebas del caso.



Figura 7.8. Tarjeta de memoria de un Samsung Galaxy

### 7.3.3 Acceso al terminal Android

A la hora de acceder al interior de un terminal Android no solo nos encontramos con la molesta presencia de ese intermediario que es el software del dispositivo, sino también con otros problemas derivados de la enorme variedad de aparatos, marcas comerciales y programas de sincronización para los diferentes terminales. En general se puede decir que es válida la misma limitación que en el iPhone: acceder a un terminal inteligente en funcionamiento, con su propio sistema operativo en ejecución y todos sus mecanismos de seguridad, no es lo mismo que realizar copias a bajo nivel de dispositivos pasivos como discos duros o llaves USB. El *smartphone* no se deja forzar con facilidad. Para obligarlo a que entregue sus datos hace falta recurrir a técnicas de *hacking* más o menos intrusivas. El grado de comprensión de jueces, fiscales y letrados de las partes contrarias puede variar en lo que respecta a este punto, pero el investigador en ningún caso debería sobreestimarlo. Más adelante veremos algunas de estas técnicas, generalmente basadas en la obtención de permisos de superusuario o *rooting* del dispositivo, y encaminadas a obtener privilegios de acceso que permiten realizar imágenes físicas del dispositivo y extraer de su interior cualquier tipo de información.

### 7.3.4 Utilidades de sincronización

Un modo inmediato y práctico de acceder lógicamente –es decir, mediante las estructuras del sistema de archivos– al terminal Android lo proporciona el propio software de sincronización que el fabricante de cada dispositivo facilita al usuario para que este pueda llevar a cabo operaciones típicas de actualización y respaldo de contactos, imágenes, documentos, etc., así como copias de respaldo de su información personal en un ordenador de sobremesa. Se trataría de un procedimiento similar al aplicado en apartados anteriores para extraer datos de un iPhone mediante los *backups* de iTunes e *iPhone Backup Extractor*.

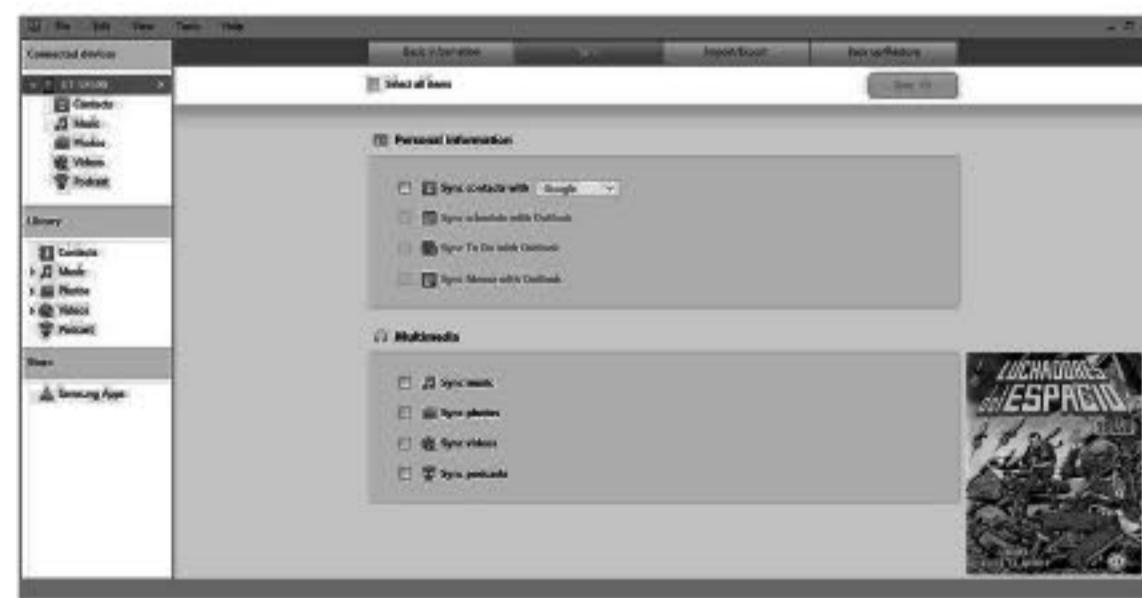


Figura 7.9. Samsung Kies

El teléfono inteligente Samsung Galaxy S2, un modelo de *smartphone* muy vendido durante los últimos meses y con prestaciones en la misma línea que los dispositivos de Apple, dispone por ejemplo de Samsung Kies PC, un software gratuito y descargable de Internet para los usuarios de cualquier terminal Samsung Galaxy 2. Una vez instalado, basta conectar el terminal al PC y configurar las aplicaciones que se quieran sincronizar o respaldar. Con Samsung Kies PC se pueden realizar copias de todos los medios –imágenes, vídeos, música, etc.– así como contactos, mensajes SMS, correo electrónico y otros contenidos de información personal. El aspecto visual del software de sincronización Samsung Kies para dispositivos con Android, como se aprecia en la figura 7.9, es similar al de Apple iTunes.



Figura 7.10. Interfaz de Android SDK

### 7.3.5 Acceso mediante Android SDK

SDK (*Software Development Kit*) es el entorno de desarrollo que la Open Handset Alliance distribuye junto con su sistema operativo para que los programadores puedan escribir aplicaciones destinadas a ejecutarse en dispositivos Android (figura 7.10). No solo dispone de compiladores y herramientas de depuración, sino también de emuladores software para algunos de los dispositivos de más amplio uso en el mercado y utilidades que permiten el acceso, tanto a los dispositivos hardware como a las máquinas virtuales que se ejecutan en los emuladores, con el objeto de comprobar el funcionamiento de las aplicaciones en un dispositivo real, así como verificar su estado, las estructuras de los sistemas de archivos y otros datos de interés. Con estas utilidades se pueden realizar copias de archivos concretos y otras tareas de análisis que, aunque pensadas en primer grado para desarrolladores y técnicos de reparación, también pueden ser utilizadas para finalidades de investigación forense.

Los entornos SDK se encuentran disponibles en versiones para Windows, Linux y OSX. Cada una de ellas tiene su propio recetario de instalación y puesta a punto que podrá encontrar en la página web del *Android Open Source Project*: <http://source.android.com/source/building-devices.html>. Una vez instalado y en funcionamiento el primer paso consiste en conseguir acceso al teléfono móvil Android mediante un cable USB. Este proceso varía en función de la versión de SDK, el sistema operativo del *host* y el dispositivo móvil que se quiere conectar, pero en general y a grandes rasgos, en la práctica totalidad de dispositivos implica los mismos ajustes básicos. En primer lugar es necesario activar el modo de depuración USB del móvil (figura 7.11) marcando la opción correspondiente en **Ajustes → Aplicaciones → Desarrollo → Depuración USB** (en inglés: **USB debugging**). Con esto lo que se consigue es dejar el terminal listo para ser utilizado como una unidad de almacenamiento de datos similar a un disco duro externo o un pendrive. Establecer la opción **USB debugging** es trivial en un dispositivo que no tenga la pantalla bloqueada por un código numérico o de trazos. De ser así habría que recurrir a alguna de las técnicas que existen para sortear este mecanismo de seguridad.

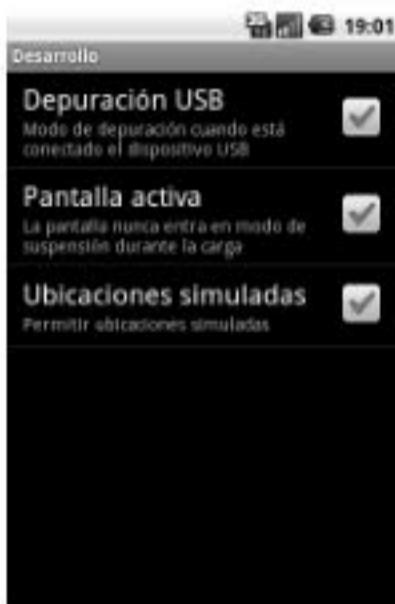


Figura 7.11. Modo de Depuración USB

Si el investigador utiliza Windows como sistema operativo en su estación de trabajo, probablemente tenga que conseguir el controlador correspondiente al teléfono móvil. Si no dispone de los CD de instalación del software de acompañamiento puede descargar el programa de sincronización del dispositivo móvil. Por ejemplo, supongamos que quiere obtener acceso a un Samsung Galaxy y no dispone del *driver*. Lo primero que tiene que hacer es descargar Kies PC, cuyo ejecutable viene empaquetado con sus librerías y datos correspondientes en un contenedor que podrá descomprimir, por ejemplo con herramientas como 7zip o rar. Dentro del archivo comprimido encontrará un directorio llamado CabFile\USB Driver, y en el interior de este el archivo SAMSUNG\_USB\_Driver\_for\_Mobile\_.rar.

Phones.exe.cab. Descomprimiéndolo a su vez con 7zip, el investigador obtendrá el ejecutable que instala los *drivers*.

En Linux el procedimiento presenta algunas diferencias. Aunque el dispositivo móvil es detectado en el arranque, no se lleva a cabo de manera automática una configuración del mismo que haga posible un acceso a través de las herramientas de SDK. Para lograr esto es necesario realizar en primer lugar algunos cambios. El investigador debe abrir una consola de texto y crear un nuevo archivo para el dispositivo USB tecleando lo siguiente con privilegios de superusuario:

```
sudo gedit /etc/udev/rules.d/99-android.rules
```

El número 99 tiene por objeto hacer que el *script* sea el último en ejecutarse, impidiendo que otros archivos modifiquen los permisos definidos. A continuación hay que agregar una línea con los datos característicos del terminal. En nuestro caso, como se trata de un dispositivo Samsung, debe introducirse el código “04e8” (si el investigador está tratando de conectar un dispositivo móvil de otra marca puede consultar la página web <http://developer.android.com/guide/developing/device.html#VendorIds>):

```
SUBSYSTEM=="usb", ATTR{idVendor}=="04e8", MODE=="0666"
```

Finalmente se guarda el archivo junto con los cambios y se reinicia UDEV mediante la orden “*sudo service udev restart*”. Si el investigador lo desea, también puede reiniciar el sistema. A partir de ese momento el terminal móvil será accesible a través de SDK y se podrán llevar a cabo en él operaciones de diversos tipos, como por ejemplo explorar el árbol de directorios, copiar archivos, etc. Una vez que se haya conseguido visualizar el árbol de directorios moverse por él resulta sencillo. Los datos del usuario y las aplicaciones instaladas desde Android Market se encuentran en el directorio /data/data.

### 7.3.6 Algunas nociones básicas de Android Debug Bridge

Android Debug Bridge comienza a funcionar en el momento de activar la depuración USB y consta de dos instancias de un mismo programa: uno de ellos se ejecuta en el dispositivo móvil y otro en el PC. Ambos actúan como servidores (adb), y para establecer comunicación con los mismos será necesario tener también un cliente adb ejecutándose en la estación de trabajo. Los servidores o *daemons* escuchan en puertos situados en el rango de 5555 a 5585. Desde el cliente en forma de consola el investigador podrá emitir un conjunto de comandos, por ejemplo para comprobar que el dispositivo está conectado:

```
forensics@ubuntu: ~$ adb devices
List of devices attached
016B756F1701200C    device
```

Otro comando útil que el investigador debe conocer es “adb shell” (figura 7.12), que permite abrir en el dispositivo móvil una consola para interactuar directamente con el sistema. En dicha consola se podrán emplear los comandos habituales del sistema operativo Linux. Esta herramienta hace posible la exploración de un sistema Android. Con ella se pueden transferir archivos y navegar por el árbol de directorios de un terminal móvil. Otras posibilidades, que pueden venir bien en caso de haberse decidido emplear alguna técnica de *rooting* para adquirir acceso privilegiado al sistema operativo del terminal con el objeto de realizar una copia a bajo nivel, incluyen la ejecución de comandos diversos en el dispositivo móvil, instalación de aplicaciones mediante la línea de comando, redireccionamiento de puertos entre el terminal y el PC, copiado recursivo de archivos y carpetas desde y al dispositivo y visualización de archivos de registro y logs.

```
C:\Windows\system32\cmd.exe - adb shell
# exit
exit
C:\Users\MatD>adb shell
# ls -l
ls -l
drwxr-xr-x root root 2010-07-01 22:22 bin
drwxrwx--- system cache 2010-06-24 10:20 cache
drwxr-x system system 2010-11-15 10:18 data
drwxr-xr-x root root 2010-11-20 10:32 dev
drwxr-xr-x root root 2010-06-30 19:17 etc
drwxr-xr-x root root 2010-11-13 22:47 host
drwxr-xr-x root root 2010-07-01 22:22 lib
drwxr-xr-x root root 2010-05-05 23:29 lost+found
drwxr-xr-x root root 2010-06-25 13:51 mnt
drwxr-xr-x root root 1970-01-01 01:00 proc
drwxr-xr-x root root 2010-06-26 00:45 root
drwxr-xr-x root root 2010-07-01 21:57 shin
drwxr-xr-x root root 1970-01-01 01:00 sys
drwxr-xr-x root root 2010-10-03 21:59 system
drwxr-xr-x root root 2010-06-26 00:44 tmp
drwxr-xr-x root root 2010-06-27 18:37 usr
-rw-r--r-- root root 118 2010-05-07 14:04 default.prop
-rwxrwxrwx root root 107412 2010-08-03 19:50 init
-rw-r--r-- root root 1714 2010-06-28 19:43 init.goldfish.rc
-rw-r--r-- root root 105 2010-06-28 19:43 init.idroid.rc
-rw-r--r-- root root 2010-06-30 01:22 linuxrc -> bin/busybox
-rw-r--r-- root root 15103 2010-08-04 20:05 init.rc
-rw-r--r-- root root 1356 2010-09-09 21:53 initDroid.sh
-rw-r--r-- root root 75 2010-06-26 08:45 init.apple.rc
lrwxrwxrwx root root 2010-11-20 10:32 d -> /sys/kernel/debug
lrwxrwxrwx root root 2010-11-20 10:32 sdcard -> /mnt/sdcard
#
```

Figura 7.12. adb shell

### 7.3.7 Significado del rooting en Android

Recuperar los archivos de un usuario sospechoso por procedimientos lógicos como los utilizados hasta el momento es importante y tiene la ventaja de que la información puede ser visualizada por aplicaciones estándar a bordo de un PC de sobremesa o un Mac. Como en el caso del iPhone, interesa en primer lugar lo que hay en los archivos gestionados por bases de datos SQLite, que es donde se encuentra casi todo lo relacionado con la actividad del usuario: listas de contactos,

llamadas telefónicas, historial de Internet, coordenadas geográficas, etc. Sin embargo, en ocasiones puede no bastar con eso. En la memoria NAND del terminal podría haber datos borrados con anterioridad, contraseñas, documentos ocultos o cualquier otro tipo de información a la que no se pueda llegar mediante las técnicas lógicas de acceso habituales basadas en la sincronización por software o el acceso directo a través de SDK. En estos casos podría ser necesario adquirir físicamente el dispositivo mediante dd u otra herramienta para la realización de copias a bajo nivel. Para ello antes el investigador deberá conseguir acceso a la memoria del dispositivo mediante una operación de “rooteado” del terminal, como la denominan los *hackers* de Android.

Aunque normalmente se le suele asimilar al *jailbreaking* de iOS, el *rooting* no tiene el mismo significado que las prácticas relacionadas con la liberación de terminales Apple y hasta hace poco consideradas delictivas. Para empezar no implica la sobreescritura de la partición del sistema con una versión alterada de la misma, sino tan solo una adquisición de privilegios similares a los de un administrador de sistemas Linux y acceso al directorio raíz (/) del dispositivo –por eso se llama *rooting*–. Una vez conseguido lo anterior se podrá realizar una imagen a bajo nivel del dispositivo, junto con los archivos borrados y el espacio no asignado por el sistema de archivos.

El procedimiento para obtener privilegios de superusuario es específico de cada dispositivo y versión del sistema. Por ejemplo, en dispositivos HTC Hero con Android 1.5 es necesario transferir al terminal un programa de terceros llamado AsRoot2 (bajado de <http://zenthought.org>). Para ello, una vez instalado y configurado SDK, deberán teclearse los comandos siguientes en un *shell* adb:

```
> adb push asroot2 /data/local
> adb shell chmod 0755 /data/local/asroot2
> adb shell
```

Con lo anterior hemos transferido al terminal la utilidad de terceros y otorgado a la misma permisos de ejecución. Acto seguido ejecutamos asroot2 para obtener privilegios de usuario. Unos pocos ajustes adicionales y la operación de *rooting* estará completa:

```
$ /data/local/asroot2 /system/bin/sh
# mount -o remount,rw -t yaffs /dev/block/mtdblock3 /system
# cd /system/bin
# cat sh>su
# chmod 4755 su
```

Tras haber ejecutado todos estos pasos de manera correcta el usuario debería tener acceso pleno a todos los recursos del dispositivo móvil, del mismo

modo que un administrador de sistemas los tiene a su máquina Linux. El proceso, como ya se ha dicho, difiere de unos dispositivos a otros según la tecnología del hardware y la versión de Android. Si el investigador tiene el propósito de adquirir privilegios de superusuario para acceder a un Samsung Galaxy o a un Nexus debe consultar la documentación técnica disponible e informarse en alguna página fiable de Internet. En cualquier caso, desde la función del superusuario o *root* podrá ver todos los archivos del sistema, modificar el software y en general hacer lo que quiera con todo lo que hay en el interior del dispositivo. Por este motivo, y también por el riesgo de dañar el terminal o los datos, durante una operación de *rooting* conviene extremar las precauciones y fijarse bien en lo que se está tecleando.

### 7.3.8 Adquisición física mediante dd

El sistema de archivos de Android se encuentra almacenado en varias ubicaciones del directorio /dev. Una vez “rooteado” el dispositivo se podrá acceder a las mismas sin ninguna restricción. Al no utilizar un disco duro convencional el *kernel* de Linux se sirve de un interfaz MTD que permite al sistema operativo del terminal ejecutarse directamente sobre una memoria *flash*. Tampoco hay reglas fijas para la designación de los archivos, y los nombres pueden variar según se trate de un modelo de teléfono u otro. Por lo general, el investigador se encontrará con algunos archivos de interés: mtd0, mtd1, mtd2, etc. Para adquirir todo el software del dispositivo basta con realizar imágenes de cada uno de estos archivos con dd. La mayor parte de la información relevante a efectos forenses se encuentra en mtd3 (archivos del sistema operativo) y mtd5 (datos del usuario).

Para realizar las imágenes se requieren nuevamente los servicios de SDK. Como antes, y después de situarse en el directorio que contiene los ejecutables, el investigador pondrá en marcha el *shell* adb y a continuación abrirá una consola con privilegios de usuario mediante la utilidad de terceros asroot2:

```
/data/local/asroot2 /system/bin/sh
```

Después de haber hecho esto podrá utilizar el comando dd igual que si estuviera adquiriendo soportes de datos en cualquier entorno Linux:

```
dd if=/dev/mtd/mtd3 of=/sdcard/mtd3.dd
dd if=/dev/mtd/mtd5 of=/sdcard/mtd5.dd
```

Las órdenes anteriores sirven para crear imágenes a bajo nivel de la partición del sistema operativo y la de datos de usuario en una tarjeta de memoria –¡imprescindible asegurarse de que haya espacio suficiente!– insertada en el *slot* del terminal. El procedimiento deberá repetirse para todas las zonas de datos accesibles a través de archivos mtdX que el investigador desee adquirir para su

análisis forense. Puesto que está trabajando con privilegios de superusuario, no estará de más insistir en que mire bien lo que hace cada vez que al utilizar dd especifica datos de origen con “if=” y de destino con “of=”. El menor descuido en esta etapa puede destruir los datos del terminal y dar al traste con la investigación.

### 7.3.9 Examen de la memoria

El estudio de las imágenes se puede llevar a cabo con las herramientas forenses habituales en busca de documentos, imágenes y medios de imagen y sonido. Sin embargo conviene hacer aquí una apreciación necesaria: muy pocas utilidades disponen de soporte para el sistema de archivos yaffs2, de desarrollo relativamente reciente y característico de los dispositivos Android. El investigador tendrá que recurrir a herramientas de *data carving* y analizar uno por uno los archivos. No obstante es posible que en un futuro esta carencia de utilidades para yaffs2 se vea paliada gracias a la creciente presencia de Android en el mercado de teléfonos inteligentes. Como poco cabe esperar que esta funcionalidad sea incorporada en herramientas forenses comerciales y alguna versión próxima de TSK.

## 7.4 RESTO DE DISPOSITIVOS Y PROCEDIMIENTOS

### 7.4.1 Supervivientes

Aunque en las páginas anteriores se ha dado prioridad a Apple iOS y Android, los dos sistemas que con el tiempo han terminado por imponerse tanto en el mercado de las comunicaciones móviles como en la cultura de masas, los dispositivos con los logotipos de la manzana y el robot no están solos. Habría sido interesante tratar sobre la investigación forense de dispositivos basados en Windows Mobile y Windows Phone, sistemas que por lo demás y según los expertos ofrecen características de seguridad empresarial mucho más sólidas que Android y Apple, aunque en el intento de seducir al consumidor no hayan tenido el mismo éxito. La presente obra por desgracia no puede extenderse a tanto. Todo ello constituye materia de libros especializados de los cuales se mencionan varios ejemplos en la bibliografía para quienes tengan interés por ampliar nociones sobre la materia.

Así mismo, aunque en relativo retroceso, están los terminales Blackberry, con una tecnología y características que también merecen capítulo aparte. Y antes que ellos fueron las agendas electrónicas, los asistentes personales y por supuesto una enorme variedad de marcas y teléfonos móviles disponibles en el mercado desde los tiempos, ya lejanos merced al efecto creado por la avalancha de innovaciones y desarrollos tecnológicos de los últimos años, en que un terminal

telefónico dejó de ser poco más que un *walkie talkie* equipado con una SIM para incorporar componentes y funcionalidad del mundo de la informática como microprocesadores, grandes cantidades de memoria RAM, tarjetas de almacenamiento de datos y en virtud de lo anterior una capacidad más que considerable para alojar en su interior archivos y elementos de evidencia interesantes a efectos de análisis forense.

Todos esos dispositivos se adquieren mediante sus propios cables propietarios y herramientas de software así mismo propietarias –algunas de las cuales se encuentran ya obsoletas, con un mantenimiento desatendido y solamente disponibles en algunas páginas especializadas de Internet–. Si alguno de estos aparatos cae en manos del investigador, muy probablemente se vea obligado a recurrir a libros en inglés de hace algunos años. Como curiosidad, y para terminar, es preciso mencionar que con algunos de ellos –como por ejemplo las agendas personales Palm en sus numerosas versiones– ya se utilizaban las técnicas de investigación forense a través de los kits de desarrollo de software o SDK.



Figura 7.13. Dispositivo UFED (CelleBrite Mobile Syncronization Ltd.)

### 7.4.2 Adquisición mediante Cellebrite UFED

Conectando el iPhone o el Samsung Galaxy a un UFED, aparato de tecnología israelí distribuido por la empresa Cellebrite (figura 7.13), se puede realizar de modo sencillo un duplicado a bajo nivel de la partición de usuario, incluyendo los archivos borrados. Originariamente el UFED (*Universal Forensic Extraction Device*: Dispositivo Universal para Adquisiciones Forenses) fue desarrollado para el sector de la electrónica de consumo, siendo sus principales clientes empresas de telefonía móvil que se servían de él para hacer duplicados de tarjetas y pasar datos de unos terminales a otros. En la actualidad lo utilizan departamentos de policía y agencias de seguridad. Viene acompañado de un maletín con cables para conectarlo a la práctica totalidad de los terminales disponibles en el mercado (teléfonos, agendas electrónicas, smartphones, tabletas, etc.). El UFED también permite realizar volcados de datos a llaves USB, tarjetas SD y otros medios de almacenamiento.

Es en la versatilidad y potencia de dispositivos como el UFED donde algunos expertos ven motivos suficientes para pronosticar que la investigación forense de dispositivos móviles tiene sus días contados. Además del UFED existe la posibilidad de adquirir y analizar teléfonos inteligentes y otros terminales móviles a través de un PC convencional mediante software como Oxygen Forensics de Danysoft y otros productos de una competencia cada vez más reñida en este sector.

## 7.5 PROCEDIMIENTOS Y RIESGOS

### 7.5.1 Alteración de las pruebas

Mientras que la adquisición y el análisis de soportes de datos convencionales como discos duros, llaves USB, disquetes de ordenador, CD/DVD-ROM y otros medios grabables y regrabables se ajusta a procedimientos estándar y generalmente reconocidos por la comunidad de investigadores y la Administración de Justicia, con los dispositivos móviles no siempre pasa lo mismo. Adquirir un disco duro a través de dd o cualquier otra herramienta a través de un bloqueador de escritura ofrece garantías plenas de que la prueba no se verá alterada durante el proceso. Si la operación se realiza de forma profesional y se cumplen todos los requerimientos en cuanto a integridad de datos y mantenimiento de la cadena de custodia, la parte contraria tendrá poco que decir.

No así en caso de que la prueba principal del caso consista en un teléfono inteligente o un *smartphone*. Algunas prácticas descritas en las páginas anteriores para la realización de imágenes forenses son claramente intrusivas y alteran los datos existentes en el dispositivo. Defender esto delante de un tribunal y frente a un abogado con experiencia en tecnologías de la información requiere como poco tener alguna razón importante que justifique el empleo de técnicas de *hacker*, así como explicar con claridad todos los pasos efectuados en la operación y dejar constancia documental adecuada de los mismos.

En realidad para verse expuesto a objeciones de este tipo no es necesario liberar el iPhone ni forzar privilegios de *root* en un terminal Android. El solo examen del dispositivo mientras está funcionando, ya sea para suspender la conectividad de red mediante el modo avión, eliminar un bloqueo de teclado o buscar información del usuario, ya implica modificaciones de algún tipo en el aparato. Es difícil interaccionar con un teléfono inteligente sin verse obligado a tomar en cada momento decisiones que pueden terminar afectando a la defensa del caso. En otras palabras, al contrario que los soportes de datos convencionales procedentes de ordenadores de sobremesa o servidores, resulta muy difícil, por no decir prácticamente imposible, examinar un dispositivo móvil sin que este acuse el

impacto de la propia actividad de análisis forense. La situación se complica en gran medida debido a la posibilidad de un borrado remoto activado por el usuario sospechoso o sus cómplices, o la ejecución automática de aplicaciones capaces de destruir información comprometedora, y que podría iniciarse en cualquier momento.

### 7.5.2 Recomendaciones ACPO

Sin embargo la presencia cada vez mayor de dispositivos móviles en las investigaciones forenses hace necesario asumir un enfoque más pragmático de esta cuestión, mediante recomendaciones oficiales, catálogos de buenas prácticas y otras medidas. Así, por ejemplo, la Asociación de Comisarios de Policía del Reino Unido (ACPO: *Association of Chief Police Officers*) ha editado recientemente una guía para la manipulación competente de medios digitales (*ACPO Good Practice Guide for Computer-Based Electronic Evidence*) que a través de cuatro puntos aspira a dar respuesta a situaciones conflictivas relacionadas con el cuestionamiento de pruebas por falta de una manipulación adecuada. He aquí los principios que rigen la operativa ACPO aplicada a todo tipo de soportes informáticos, y de modo particular a dispositivos móviles:

- Las agencias encargadas del orden público y su personal no deben emprender en ningún momento actuaciones que pudieran alterar los datos contenidos en un soporte de almacenamiento y que con posterioridad vayan a ser utilizados en un proceso judicial.
- En aquellas circunstancias en que resulte necesario acceder a los datos originales contenidos en un ordenador o un medio de almacenamiento de información, deberá hacerse a través de una persona competente y capaz de explicar la relevancia y las implicaciones de sus actos.
- Deberá existir trazabilidad y un historial auditable de todos los procesos aplicados en el tratamiento de evidencias digitales. Siguiendo este historial un perito independiente debería obtener los mismos resultados.
- La persona encargada de dirigir la investigación (*case officer*) es responsable del cumplimiento de la ley y de los principios expuestos con anterioridad.

### 7.5.3 Intervención de un dispositivo móvil

A diferencia de los PC y los soportes de datos convencionales, la intervención de dispositivos móviles –incluyendo en este apartado no solo los

estudiados con anterioridad sino también otros como agendas electrónicas, Blackberries y teléfonos celulares de cualquier marca y modelo— requiere un protocolo de actuaciones preciso al efecto no solo de proteger el material y asegurar la cadena de custodia, sino también para evitar posibles maniobras remotas del usuario que pudieran comprometer la integridad de los elementos de evidencia contenidos por el aparato.

- a) **Aislamiento de redes.** Normalmente es lo primero que hay que hacer cuando se encuentra un dispositivo móvil. Smartphones y teléfonos inteligentes incluyen una funcionalidad especial llamada “modo avión” que desactiva por completo la conectividad GSM, WiFi y Bluetooth, dejando al terminal convertido en poco menos que una agenda electrónica sin conexión con el mundo exterior. Esta característica tiene su origen en la necesidad de cumplir con la antigua prohibición de usar teléfonos móviles a bordo de aviones en vuelo. En un iPhone, por ejemplo, resulta fácil llegar hasta ella, a través de **Ajustes → Modo avión**. Cuando el dispositivo no ofrezca esta funcionalidad (como suele ser el caso en la mayor parte de los teléfonos móviles) la única manera de aislar el aparato de la red consiste en meterlo dentro de una bolsa especial con filamentos metálicos, constituyendo una jaula de Faraday que impida toda comunicación con el exterior a través de ondas electromagnéticas. El dispositivo deberá ser trasladado entonces sin pérdida de tiempo al laboratorio forense, ya que los repetidos intentos por re establecer la conexión mediante la emisión de señales a torre pueden agotar rápidamente la batería.
- b) **Anulación de códigos de protección.** En el caso de que el aparato esté protegido por un código numérico o de trazos y sea incautado en el preciso momento en que el usuario sospechoso esté haciendo uso de él, con un cierto tiempo —que puede ser de segundos a varios minutos— antes de que se active la protección, deberán cambiarse los ajustes para prolongar este tiempo o, de ser posible, eliminar la protección mediante código. En un iPhone, nuevamente, se llega hasta estas características a través del menú **Ajustes → General → Bloqueo numérico/bloqueo con código**. Este paso debe seguirse antes de aislar el dispositivo móvil de la red mediante alguno de los procedimientos indicados con anterioridad. Si el terminal estuviera bloqueado, existen técnicas especiales para adivinar o sortear los códigos. Naturalmente todo esto implica algún tipo de alteración del dispositivo y puede resultar crítico para la defensa del caso, por lo que conviene actuar precavidamente y dejando constancia documental de cada una de las actuaciones que se lleven a cabo.

- c) **Cables de alimentación y datos.** La mayor parte de los laboratorios forenses dedicados al análisis de dispositivos móviles disponen de juegos completos de cables en prevención de cualquier necesidad urgente que pueda presentarse. No obstante forma parte de una buena práctica forense hacerse cargo, si fuera posible, de los cables hallados en el lugar de los hechos. Esto puede ser necesario por ejemplo en el caso de algunos dispositivos modernos como el Dell Streak y la Samsung Galaxy Tab, ambos con Android y sincronizados a través de un interfaz de reciente diseño denominado PDMI (*Portable Digital Media Interface*), que se emplea tanto para recargar baterías como para trasladar vídeo de alta resolución mediante USB 3.0. Sería imperdonable que la adquisición y el análisis forense de uno de estos dispositivos se retrasaran o llegaran a hacerse imposibles por agotamiento de la batería mientras se intenta localizar en el comercio un cable de esas características.
- d) **Dispositivos desconectados.** Si un terminal se encuentra desconectado en el momento de su hallazgo la mejor opción consiste en arrancarlo en modo recuperación para comprobar la conectividad y —en el caso de Android— las posibilidades de acceder a él en modo *root*. Encender un teléfono inteligente solo para comprobar qué sistema operativo lleva no es una buena idea. En caso de que sea necesario conectarlo para proceder a su análisis, el investigador debe tener en cuenta todo lo comentado antes acerca de la necesidad de explicar y documentar cada uno de sus pasos.



Figura 7.14. ¡No olvide poner el dispositivo en modo Avión!

### 7.5.4 Riesgo legal

Probablemente el lector se haya quedado con la impresión de que este capítulo es incompleto o insuficiente para sus necesidades informativas u operativas sobre el tema. Se trata sin embargo de algo inevitable. Mientras que la adquisición y el análisis forense de ordenadores de sobremesa y soportes de datos convencionales admiten métodos estándar y consolidados desde hace años, todo lo que tiene que ver con los dispositivos móviles reviste un carácter fundamental de provisionalidad. No sabría decirle cuánto de lo que ha leído seguirá siendo válido de aquí a unos pocos meses. No obstante, más que con unos procedimientos y una tecnología que pueden cambiar de la noche a la mañana, me gustaría que el lector se quedase con otras nociones que le van a resultar más útiles en caso de que decida dedicarse a la investigación forense de dispositivos móviles, y que paso a exponerle a continuación.

La mentalidad optimista y lineal del ingeniero tiende a ver el mayor obstáculo en las dificultades técnicas que se van presentando durante el trabajo. Pero aquellas no son nada comparadas con el campo de minas legal en que se desarrolla la actividad de un investigador forense. Muchas veces, y como resultado de las desagradables experiencias obtenidas al respecto, se tiene la impresión de que el enemigo no es la naturaleza, ni siquiera el delincuente, sino un entorno jurídico que, al haber vulnerado una oscura norma de cuya existencia no se tenía ni la menor idea, de repente se vuelve contra un investigador que actúa de buena fe o un fiscal que simplemente se limita a hacer su trabajo. Argumentos como este son frecuentes: “el perito realiza divagaciones caprichosas”, “con su intervención en los ordenadores de la red provocó daños en la infraestructura informática de mi cliente”; peor aún, “manipuló la evidencia”, o, aunque no lo haya hecho, “olvidó poner el bloqueador de escritura”. El caso extremo, como ya ha pasado en más de una ocasión, es el de abogados de la defensa que con el objeto de confundir al perito plantean preguntas absurdas o desconcertantes como por ejemplo: “¿puede usted demostrar la existencia del protocolo TCP/IP?”

### 7.5.5 Privacidad

Estos peligros se vuelven más patentes cuando se trata de la investigación forense de dispositivos móviles. Por lo general las objeciones a las que antes se ha hecho mención se emplean con el propósito de anular informes o alcanzar acuerdos. Mucho más grave, sin embargo, resulta oír argumentos del tipo: “¿y quién le dio a usted permiso para curiosear en la información personal de mi cliente?”. Los teléfonos móviles inteligentes, *smartphones* y aparatos similares han terminado por integrarse en el ámbito individual hasta el punto de terminar constituyendo un reflejo revelador de la vida de sus propietarios: identidad personal, correspondencia íntima, movimientos por la ciudad, aficiones, problemas

de salud, saldos bancarios, infidelidades y otros pequeños secretos. Todo esto, que está tan a la mano como las hojas de los balances o el historial de un servidor web, adquiere un tremendo potencial explosivo en el momento en que se cruza el umbral de la Administración de Justicia.

La Constitución protege el derecho a la privacidad como uno de sus bienes preferentes. Lo mismo hacen otras leyes de rango superior basadas en normas fundamentales del Estado de Derecho. Para convencerse de lo efectiva que aspira a ser esta protección, así como del empeño de los juristas en velar porque ello siga siendo así, el lector no tiene más que echar un vistazo a las resoluciones de la Agencia Española de Protección de Datos, en las que se ofrece un historial completo de casos, fundamentaciones jurídicas y sanciones aplicadas. La idea de que cuatro cables y un par de libros en inglés son material suficiente para establecerse como analista de dispositivos móviles para detectives privados o maridos celosos es peligrosamente frívola, y todas esas pequeñas empresas que se están estableciendo para dedicarse a la explotación de lo que piensan puede ser un lucrativo nicho de negocio no tardarán en comprobarlo en cuanto un fiscal las ponga en su punto de mira. Si tuviera que quedarse con algunas ideas que resuman todo lo explicado en el presente capítulo en relación con los riesgos y dificultades de la investigación forense de teléfonos inteligentes y dispositivos móviles, el autor preferiría que sean precisamente estas.

## Capítulo 8

# INVESTIGACIÓN DE IMÁGENES DIGITALES

Imagine la siguiente situación: la policía acaba de incautar un *smartphone* Android bloqueado con patrón de pantalla. Para acceder a él es necesario aplicar alguna de las diversas técnicas de desbloqueo –inevitablemente intrusivas– utilizadas por especialistas y *hackers*. Nos podría ahorrar trabajo conocer el patrón de desbloqueo, es decir, la forma del trazo que el usuario tiene que hacer con su dedo sobre la pantalla del dispositivo para que este le permita acceder a su interfaz principal. El agente de policía saca con su cámara digital una foto del teléfono móvil con la pantalla apagada y la descarga en su ordenador. Luego la abre con Photoshop, modifica los ajustes de luminosidad y contraste, aplicando además otras técnicas de realce hasta que sobre la imagen se vuelven visibles las marcas dejadas por el dedo del usuario. De este modo obtiene acceso al dispositivo de modo rápido y sencillo y sin necesidad de aplicar ninguna técnica intrusiva.

Esto es posible debido a la forma en que las deposiciones de grasa corporal, sudor y pequeñas partículas de suciedad ambiental alteran las propiedades reflectantes de la pantalla durante el uso del dispositivo. En general, y como demuestra un estudio realizado hace pocos años por el departamento de Ciencias de la Computación de la Universidad de Pennsylvania (Aviv, Gibson, Mossop, Blaze, Smith: *Smudge Attacks on Smartphone Touch Screens*) una pantalla limpia es principalmente reflectiva y en menor grado difusa; pero cuando está sucia la difusividad predomina sobre la reflectividad. Mediante iluminación oblicua se puede obtener una imagen de la superficie que a través de unos pocos ajustes mostraría diferencias de contraste reveladoras de la imagen del patrón.

Si los responsables de la intervención toman precauciones para minimizar el contacto manual con el dispositivo esta técnica puede ser aplicada en la mayor parte de los casos. A no ser, claro está, que el sospechoso sea un individuo extremadamente aseado y pulcro. En cualquier caso la lectura de ese documento de la Universidad de Pennsylvania es altamente recomendable por ofrecer el estudio más completo realizado hasta la fecha sobre la problemática del *smudge attack* o ataque a pantallas sin limpiar. Al mismo tiempo hemos puesto un ejemplo de cómo una sencilla técnica de proceso de imagen puede ser de gran ayuda para la investigación forense.



Figura 8.1. Smudge attack (cortesía de Aviv, Gibson, Mossop, Blaze y Smith)

## 8.1 INFORMÁTICA FORENSE E IMÁGENES DIGITALES

El ser humano vive en un mundo visual. La conciencia de este hecho, presente en forma de numerosos estudios sobre arte, color y perspectiva realizados desde los tiempos de Leonardo da Vinci hasta el presente, se manifiesta de modo masivo desde hace unos pocos años con la irrupción de la informática en la cultura popular. La digitalización de la imagen constituye un ejemplo del cambio de paradigma citado en la introducción del libro, desde tecnologías capaces de representar la realidad a otras que nos permiten manipularla en formas que hasta hace poco habrían resultado inimaginables. En la actualidad se puede decir que no existe área de la técnica o la economía en la que no se apliquen técnicas de proceso digital de imagen: investigación científica, publicaciones de revistas y libros, mediciones industriales, vigilancia y seguridad, entretenimiento, creación artística, etc. La generalización de la cámara digital para mercados de consumo e Internet como fenómeno de masas ha convertido el mundo en un enorme caleidoscopio.

Esta inundación de imágenes liberadas del soporte fotoquímico y compuestas en última instancia por unos y ceros, ubicuas, manipulables y reproducibles hasta el infinito sin pérdida de calidad influye inevitablemente en la actividad del investigador. Sin embargo la relación entre el mundo de la imagen

digital y la Informática Forense es más compleja de lo que parece. Cuando se habla de investigación de imágenes digitales, la acepción inmediata del término tiene que ver con actividades de análisis encaminadas a averiguar si una imagen es auténtica o producto de una manipulación con Adobe Photoshop, GIMP o cualquier otro software de retoque fotográfico. Aparte de esto la fotografía digital, con toda su batería de procesos informáticos para tareas de ajuste y realce de imagen, también es utilizada como herramienta de investigación con el objeto de aportar pruebas y documentar informes periciales. Y para terminar está el análisis forense de los metadatos de archivos gráficos. A lo largo de las páginas siguientes vamos a pasar revista a estas tres acepciones de la investigación forense de imágenes digitales.

## 8.2 IMÁGENES MANIPULADAS

### 8.2.1 ¿Verdadero o falso?

El hecho de que las imágenes gráficas tengan una presencia generalizada en el mundo de hoy hace que por el mismo hecho de su abundancia y ubicuidad pase desapercibida su naturaleza como bien económico susceptible de operaciones ilícitas como robo de datos, utilización indebida o falsificación: investigaciones científicas dotadas con becas millonarias, fotografías acreedoras a un premio Pulitzer, imágenes trucadas de famosos o personajes de la política, fotos de modelos de alta costura que con inalcanzables cánones de belleza fomentan trastornos psicológicos de la alimentación. En una escala menos espectacular pero no por ello menos generalizada y dañina existe toda una picaresca basada en la utilización de software de retoque fotográfico para manipulaciones de todo tipo: estafa al seguro mediante abolladuras fingidas y cambios de matrícula de los vehículos, falsificación de títulos académicos y documentos oficiales, venta de fotografías antiguas en eBay y un largo etcétera. Los motivos para investigar la autenticidad de imágenes digitales van por consiguiente más allá de lo anecdótico y atienden a diversos fines, desde el esclarecimiento de crímenes de estado hasta casos de *mobbing* y gamberradas de instituto.



Figura 8.2. Fotografía retocada (izq.) de los bombardeos de Beirut (REUTERS/Adnan Hajj)

Observe la figura 8.2. Se trata de una imagen de los bombardeos de Beirut por el ejército israelí en el año 2006 tomada por el fotógrafo Adnan Hajj, un *freelance* libanés de la agencia Reuters. La fotografía tuvo que ser retirada cuando el público comenzó a protestar por las manipulaciones introducidas en la misma (imagen izquierda) con el objeto de realzar el dramatismo de los ataques. En este caso las manipulaciones son perceptibles a simple vista, y consisten en una intensificación del contraste así como en el empleo de partes clonadas procedentes de la zona más densa de la humareda.

Dada la considerable influencia que a veces una imagen puede tener sobre la sociedad conviene disponer de métodos que permitan verificar la autenticidad de la misma. La manipulación de fotografías siempre ha existido, y algunas de las técnicas tradicionales utilizadas no resultan desconocidas para el lector: dobles exposiciones, montaje de negativos, inserción de máscaras de cartón, retoque de negativos con pincel, etc. El resultado eran esas fotos de Stalin purgadas de adversarios políticos, en las que un apoyabrazos de butaca pintarrajeado o la diferente dirección de las sombras ponían al descubierto el engaño incluso ante los ojos de un observador poco experimentado.

Tergiversar la realidad con tecnologías analógicas resultaba algo complicado y costoso. Para dar el pego se necesitaba invertir una cantidad considerable de tiempo y de talento artístico. En la actualidad, sin embargo, las herramientas de retoque fotográfico como Adobe Photoshop o GIMP permiten realizar unos montajes que habrían sido la envidia de los dictadores de antaño. La ausencia de negativos fotoquímicos y la posibilidad de aplicar procesos ecualizadores de luminosidad y color en la imagen hacen que en ocasiones resulte extremadamente difícil determinar si una fotografía es auténtica o falsa. ¿De qué recursos puede servirse en tales circunstancias el investigador?

### 8.2.2 ¿Cómo funciona una cámara digital?

Antes de hablar sobre los métodos y las herramientas informáticas empleadas en el análisis forense de imágenes digitales es preciso entender cómo funciona una cámara digital. La mayor parte de los modelos que se venden en la actualidad disponen de un CCD o un chip CMOS, que en lo más básico viene a ser como una especie de cuadrícula formada por elementos fotosensibles capaces de desprender cargas eléctricas al ser bombardeada por fotones procedentes del exterior a través del objetivo de la cámara. Los electrones saltan, cada uno desde su celda fotosensible y en número proporcional a la intensidad de luz generada por ese punto concreto de la escena, y caen en un sumidero del cual son extraídos a un registro de desplazamiento que los transporta por hileras a un conversor analógico-digital (AD). Finalmente son almacenados de manera ordenada en un chip de

memoria formando el archivo gráfico en bruto (RAW) de la imagen. Posteriormente una lógica integrada en la propia cámara o un software especial los convierte en los archivos JPG o PNG que el usuario visualiza y procesa en su ordenador por medio de aplicaciones convencionales.

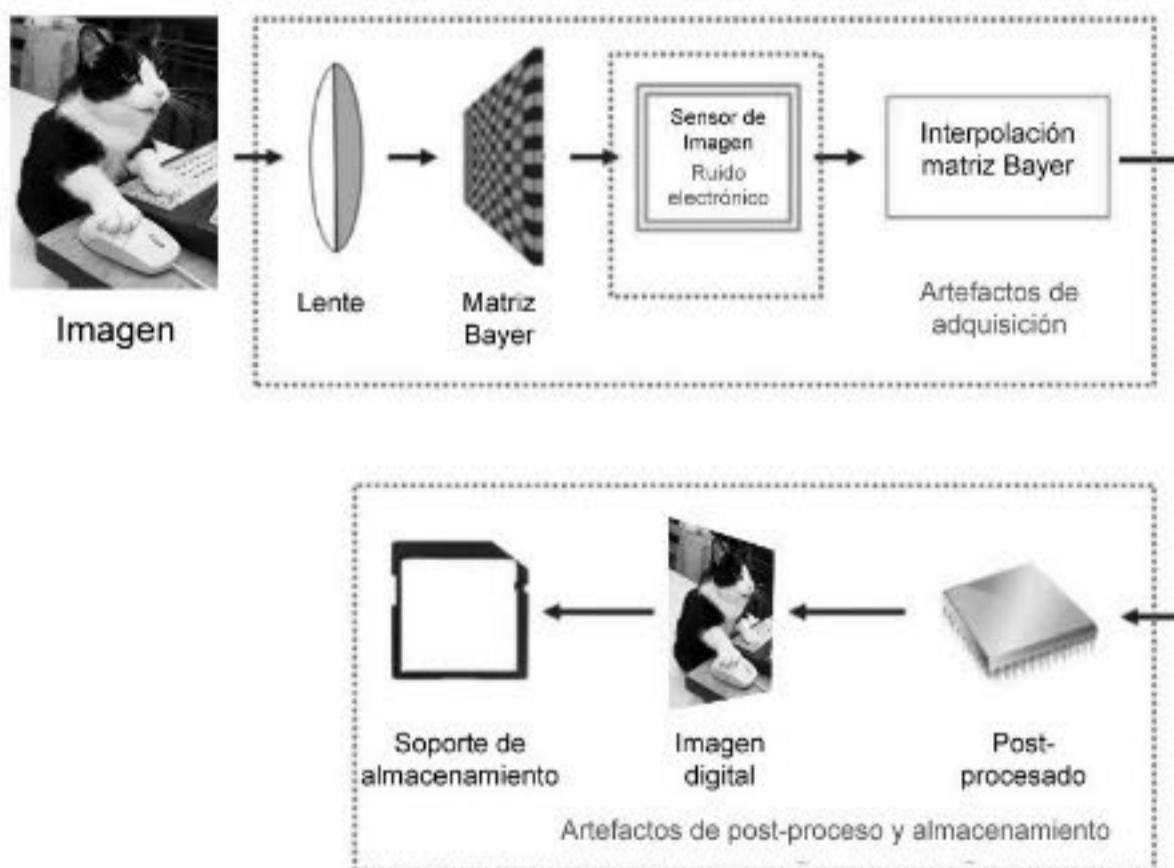


Figura 8.3. Esquema de funcionamiento de una cámara digital

Un lector avisado puede estar preguntándose: ¿y de dónde viene el color? Porque el procedimiento descrito únicamente permite obtener fotografías en blanco y negro. Lo mismo que el conversor AD solo es capaz de transformar una señal en una ristra de números, sin preocuparse para nada de si lo que está procesando son imágenes, sonido, temperatura o cualquier otra señal cuyo valor experimenta variaciones dentro de un rango continuo, los elementos fotosensibles del CCD solo entienden de intensidad de luz. Para lograr un efecto cromático se superpone al chip una matriz de celdas coloreadas que actúan como filtro. Existen diversos diseños para estas matrices. El más conocido es el patrón Bayer –inventado por un ingeniero de Kodak y no por la multinacional química alemana–, una cuadrícula compuesta por celdas verdes, rojas y azules en proporción definida de acuerdo con un modelo psicofísico de percepción del color por el ojo humano (motivo por el cual en la matriz Bayer la mitad de las celdas corresponden al color verde).

A parte de la matriz Bayer existen otros diseños de cuadrícula e incluso sistemas con triple matriz de color, en los que la luz incide tras haber sido fragmentada convenientemente por un complicado y caro montaje de prismas, destinados a cámaras de gama alta o aplicaciones especiales. El sistema que se

acaba de describir es el más extendido y se utiliza en la mayor parte de las cámaras del mercado. Las consideraciones que se exponen a continuación se refieren a la cuadrícula Bayer y otras de tipo similar, en las que los tres colores de la mezcla aditiva –rojo, verde y azul– se hallan dispuestos en una sola capa.

### 8.2.3 Interpolación e inconsistencia estadística

En este diseño característico basado en la matriz Bayer solo hay un filtro y no tres, como sería lógico para obtener una imagen compuesta por una mezcla de los tres colores básicos. Cuando la cámara digital realiza una toma el resultado es un mosaico de píxeles alternantes de color, muy parecido a los cuadros de los pintores puntillistas del impresionismo francés del siglo XIX. Para crear una imagen cromática con zonas continuas el software de la cámara –o la lógica integrada en el hardware según el caso– lleva a cabo una redistribución del color, recalculando el valor de cada píxel o tesela del mosaico en función del valor de las celdas adyacentes. Este proceso se denomina “interpolación” y para conseguirla existen diversos algoritmos matemáticos. Por ejemplo se puede establecer el valor de cada píxel promediando el de sus dos vecinos adyacentes (interpolación lineal) o bien obteniendo el promedio de las 16 celdas próximas (interpolación bicúbica).

Todo esto parecerá extraño a quien haya adquirido sus primeras nociones de fotografía con la técnica fotoquímica de hace tan solo unos pocos años, basada en la elaboración del color a base de filtros superpuestos. Sin embargo el resultado final de todas estas operaciones es una imagen fotográfica de rango tonal completo que, dicho sea de paso, gracias al procedimiento de su elaboración proporciona al investigador forense un método que puede resultar útil a la hora de decidir si una imagen digital ha sido manipulada.

El que la imagen digital se obtenga por métodos de interpolación implica que los valores adyacentes de sus píxeles están correlacionados de algún modo por efecto de las operaciones matemáticas llevadas a cabo por el software de interpolación. Si en alguna zona de la imagen esta correlación deja de existir, ello apuntaría a la existencia de algún tipo de manipulación mediante añadidura de elementos procedentes de otras imágenes, cambios directos en los píxeles u operaciones de transformación de cualquier otro tipo realizadas a través de un software de retoque fotográfico.

### 8.2.4 Artefactos

Con frecuencia, al examinar la imagen ampliada en un monitor, estas alteraciones se perciben a simple vista en forma de aberraciones cromáticas o transiciones extrañas en los contornos de las figuras (*artifacts*). De hecho uno de

los principios en que se basa la detección de imágenes falsas tiene que ver con la presencia de dichas anomalías: el análisis de inconsistencias estadísticas en archivos gráficos. Los artefactos no solo se crean mediante la manipulación intencionada de la imagen digital, sino que también son el resultado inevitable de la forma en que funcionan las cámaras digitales. Todo proceso electrónico o de computación genera defectos característicos que alteran la calidad de la imagen o modifican de alguna manera sus propiedades.

Así, por ejemplo, el sensor de la cámara produce ruido por efecto del choque aleatorio de electrones en el interior de sus circuitos. El proceso de interpolación también genera sus imperfecciones que son como una especie de huella digital. Finalmente las tareas de post-procesado de la imagen, necesarias al efecto de convertir en un archivo de datos el caudal de señales digitalizadas emitido por la electrónica del sensor, disponen de sus artefactos correspondientes. Comprender el origen y las características de los diferentes tipos de artefactos es importante a la hora de buscar indicios que apuntan a la posible manipulación de una imagen digital. En la figura 8.3 se puede apreciar la dependencia de los distintos tipos de artefactos con respecto a los subsistemas funcionales que los generan en el proceso de elaboración de una fotografía digital.

### 8.2.5 Zonas clonadas

Una manipulación habitual en el retoque fotográfico consiste en copiar una zona de la imagen para copiarla sobre otra, sobreescritiendo la información de los píxeles originales. Los motivos para la clonación pueden ser variados. Normalmente se trata de suprimir en la imagen original algo que no interesa que se vea. Otras veces el objetivo del falsificador consiste en realzar una fotografía original mediante la repetición de alguna parte determinada que dé énfasis a la escena. Los bombardeos en Beirut (figura 8.2) resultan más dramáticos salpicando la humareda con pequeños nubarrones negros copiados a partir de la zona más oscura de la columna de polvo procedente de los edificios dañados. En estos casos los intentos de manipulación son tan burdos que su detección resulta fácil incluso para un observador poco experimentado. Otros fotógrafos diestros con el Photoshop, sin embargo, son más hábiles a la hora de crear imágenes de gran impacto o crear falsos resultados en la documentación gráfica de experimentos científicos.

El clonado ofrece buenas perspectivas de éxito cuando se realiza sobre áreas continuas como un fondo de color y textura uniformes, como por ejemplo praderas de hierba, un bosque, agua o cielo despejado. La razón para clonar una extensión de pradera o de playa sería por ejemplo el deseo de extenderla por encima de un objeto determinado para evitar que se vea en la imagen. En tales circunstancias la comparación no puede realizarse a ojo sino con la ayuda de una

herramienta informática capaz de detectar repeticiones de valores en determinadas zonas de la imagen formadas por piíxels adyacentes.

Existen algoritmos matemáticos adecuados para esta tarea. Como la comparación de áreas píxel por píxel resulta un proceso extremadamente intensivo en cálculo y en ciclos de procesador, el modo de funcionamiento de dichas aplicaciones consiste en dividir la escena en bloques de un número determinado de píxeles, promediar sus valores y tratar de localizar equivalencias en otras partes de la imagen. La búsqueda de zonas clonadas, al igual que la de manipulaciones conseguidas a base de retoque fotográfico o insertar material gráfico procedente de otras imágenes, también se basa en el análisis de correlaciones estadísticas. Pero en este caso lo que se busca no es la diferencia con respecto a la correlación normal en un grupo de píxeles adyacentes, sino una repetición de las mismas correlaciones matemáticas en varias zonas de la imagen.



Figura 8.4. Esta fotografía es falsa

### 8.2.6 Inconsistencias en la iluminación

Este es un criterio forense que antes de la era digital se utilizaba ya para detectar falsificaciones fotográficas. Si en una imagen dos objetos no reflejan la misma cantidad de luz, y si además los mismos parecen tener diferentes fuentes de iluminación, lo más probable es que la fotografía esté manipulada. Esto no solamente es válido para el controvertido álbum de familia de la Revolución Rusa, donde hay algunas fotografías de Stalin en compañía de un Lenin que al parecer

nunca estuvo donde la imagen le muestra (figura 8.4), sino que fue añadido a posteriori por conveniencias de la propaganda: la inconsistencia en la iluminación sirve de igual modo para detectar falsificaciones en imágenes digitales (figura 8.5). Hallar en dos fotografías diferentes objetos que tengan las mismas condiciones de iluminación es muy difícil, y aunque podamos promediar el brillo y el contraste, resulta imposible procesar una de las imágenes de manera que las sombras proyectadas sugieran que la luz que incide sobre ella procede de la misma fuente que ilumina a la otra.



Figura 8.5. Esta otra también (originales sin alterar propiedad de Ken Light y Franken/CORBIS)

Existen innumerables ejemplos de este tipo de manipulación, y seguramente no pasará un día sin que el investigador descubra en los medios o en Internet algún especimen nuevo que llame su atención. Merece la pena mencionar la famosa portada de la revista *Life* (febrero de 1964) en la que aparece Lee Harvey Oswald posando en el patio de su casa con un periódico marxista y el rifle Mannlicher-Carcano que presuntamente utilizó para asesinar al Presidente Kennedy. Todavía hoy numerosos observadores, basándose en la percepción intuitiva de algunas incoherencias en las sombras y las dimensiones corporales de Oswald, sostienen que se trata de un montaje hecho con fotografías distintas. Una reciente investigación realizada por el Dr. Hani Farid, experto en análisis forense de imágenes digitales del Dartmouth College, mediante modelos 3D compuestos a partir de los elementos de la imagen, parece haber dilucidado todas estas dudas al determinar la existencia de una fuente de luz única y coherente. La foto de Oswald con fusil y panfleto en el patio de su casa es a todas luces –tanto en sentido figurado como literal– auténtica. Si esa portada de *Life* fuese a pesar de todo una manipulación, cuesta creer que alguien la hubiera podido fabricar con los medios técnicos disponibles en el año 1963.



Figura 8.6. ...pero esta, al parecer, es auténtica (Fuente: Wikipedia Commons)

Las inconsistencias en la iluminación no solo afectan a las sombras proyectadas por los objetos, sino que se pueden percibir en los reflejos de la luz sobre superficies brillantes, y en determinados casos, sobre los ojos de las personas retratadas. El propio Dr. Farid, que es la mayor autoridad mundial en el campo de la investigación de imágenes digitales, ha desarrollado herramientas de software que permiten determinar la posición de fuentes de luz a partir de los reflejos que las mismas producen en la parte transparente del globo ocular. Este, debido a su forma redonda, proporciona una valiosa información sobre el emplazamiento de las fuentes de luz incidentes, tanto puntuales como difusas, haciendo posible la detección de incoherencias que sugieran que la imagen ha podido ser manipulada.

### 8.2.7 E.L.A. (Error Level Analysis)

Otro procedimiento para determinar la autenticidad o falsedad de una imagen digital consiste en analizar la distribución de errores estadísticos resultantes del proceso de compresión con pérdida de los archivos gráficos. Este método es aplicable a imágenes JPG, principal formato con el que trabajan las cámaras digitales por razones sobre todo de economía, ya que ofrece una excelente relación calidad-tamaño así como la posibilidad de elegir entre diferentes niveles de compresión para aprovechar mejor el espacio de los soportes de datos y facilitar la carga por los navegadores de Internet.

Prescindiendo de la teoría matemática, el método puede definirse así: cada vez que una imagen JPEG se guarda con un ajuste menor del 100%, es decir, con pérdida, se producen errores que aunque no perjudiquen a la calidad percibida por el ojo pueden medirse matemáticamente a través de un algoritmo. En operaciones sucesivas de guardado del archivo estos errores se van acumulando, de modo que si

en un momento determinado alguien modifica el archivo con un software de retoque fotográfico, realizando ajustes en partes concretas de la escena o pegando áreas recortadas desde otras imágenes, los niveles de error correspondientes a esas zonas nuevas serían distintos a los del resto de la imagen. La existencia de una falta de correlación estadística en este sentido podría admitirse como prueba de que la imagen ha sido manipulada.



*Figura 8.7. Error Level Analysis*

En la práctica la diferencia se muestra mediante cambios de color o tonalidad. Las zonas claras de la imagen tienen un nivel de error más bajo, de lo cual cabe deducir que no forman parte de la fotografía original, sino que fueron incorporadas posteriormente. Los métodos basados en el análisis de niveles de error han dado resultados espectaculares con fotografías procedentes de páginas web del mundo de la moda, que por abundar en un manejo desvergonzadamente imaginativo y a veces poco ético de los programas de retoque fotográfico constituye un banco de pruebas ideal para las herramientas E.L.A. Queda por ver si podrían demostrar su eficacia también en los tribunales.

### 8.3 UTILIZACIÓN COMO HERRAMIENTA FORENSE

El uso de la imagen digital como medio de investigación y consecución de pruebas periciales nos coloca en una situación diferente. En los apartados anteriores se ha pasado revista a diversas técnicas que ayudan al investigador forense a detectar manipulaciones en una imagen digital con la idea de que un jurista la pueda impugnar (o defender en caso de probarse su autenticidad) delante de un tribunal. Ahora de lo que se trata es de obtener imágenes solventes y de alta calidad que no puedan ser atacadas con éxito por la parte contraria o rechazadas por el juez. Para este propósito, además de disponer de un catálogo de buenas prácticas y de los medios precisos, el investigador deberá hallarse al tanto de la

legislación en vigor y el tratamiento aplicado de la prueba electrónica, así como la normativa aplicable a fotografías digitales si la hay, en el ordenamiento jurídico dentro del cual lleva a cabo su labor como experto forense.

#### 8.3.1 La imagen digital como prueba

Aunque el uso de la fotografía como medio de examen pericial y de presentación de pruebas es antiguo, el ordenamiento jurídico español, basado fundamentalmente en leyes procesales del siglo XIX, no dispone de una doctrina específica y unitaria que regule la utilización de elementos de evidencia gráficos. La situación se complica con la fotografía digital, que a su carácter novedoso añade problemas jurídicos derivados de su naturaleza informática, con las posibilidades ilimitadas de copia y manipulación que la misma conlleva. Lo mismo se puede decir en general de los archivos de datos guardados en soportes magnéticos, ópticos o de otros tipos.

Generalmente, y a falta de una opinión consolidada, en la normativa que regula el tratamiento de pruebas judiciales las imágenes digitales se asimilan al documento electrónico, el cual es considerado por los jueces como una variante especial del documento como elemento probatorio. Por documento, a efectos procesales y en una definición que goza de amplio apoyo en la jurisprudencia, se entiende “todo objeto material representativo de un hecho de interés para el proceso, representación que puede obtenerse (...) mediante los modernos medios reproductivos, como la fotografía, la fonografía, la cinematografía, el magnetófono, las cintas de vídeo, los discos de ordenador y cualesquiera otros similares” (Serra Domínguez, M.). De acuerdo con lo anterior una fotografía digital es un documento y ha de recibir el tratamiento correspondiente ante un tribunal.

En cuanto a dicho tratamiento, según el artículo 319 de la Ley de Enjuiciamiento Civil, la validez de un documento público no necesita ser probada. El hecho de haber sido emitido por fedatario público o por un agente autorizado de la Administración lo convierte automáticamente en un instrumento válido ante el juez. En todo caso quien alega su falsedad debe promover el correspondiente procedimiento penal para conseguir que sea excluido y no produzca efectos probatorios. Con respecto a los documentos privados, el artículo 326 de la misma Ley de Enjuiciamiento Civil establece que tendrán validez en los mismos términos que señala el artículo 319 para los documentos públicos, toda vez que su autenticidad no sea impugnada por la parte a quien perjudiquen. En otras palabras, si la parte contraria impugna la autenticidad del documento, la parte que lo presenta deberá aportar pruebas de que efectivamente lo es.

Si la fotografía ha sido obtenida en presencia de un notario y este levanta acta del procedimiento, en principio la prueba es válida y el juez la admitirá sin más trámite, a no ser que el letrado de la parte contraria solicite los servicios de un perito lo suficientemente hábil para impugnarla. Si la imagen es adquirida por un experto este deberá seguir un método de trabajo diáfano y bien documentado que permita defender su prueba delante de un tribunal con la necesaria solvencia profesional para impedir que la parte contraria la impugne. En ninguna de las dos situaciones está garantizado el éxito, ya que existen numerosos factores que pueden intervenir en el proceso y además el art. 382.3 de la LEC establece como regla general que las pruebas han de ser evaluadas conforme a las reglas de la sana crítica. Es decir, de un modo estricto equipara esta valoración de prueba a una declaración de testigos, porque ambas se valoran conforme a la sana crítica, perteneciendo por tanto, al orden subjetivo del entendimiento del Juez, de acuerdo a interpretaciones objetivas extraídas de las “máximas de la experiencia” y de su propio conocimiento.

### 8.3.2 Recomendaciones SWGIT

La necesidad de aplicar procedimientos diáfanos afecta no solo al procesado de la imagen sino a todo el acto probatorio relacionado con la fotografía desde el momento en que se aprieta el botón del disparador hasta la defensa delante del tribunal. Ha de tenerse en cuenta además que la fotografía no es una técnica de representación de la realidad enteramente imparcial. Aunque no se introduzcan en ella modificaciones ni retoques de ningún tipo existen formas sutiles de condicionar su influencia sobre el público al cual las imágenes van destinadas, en este caso los magistrados o un jurado. El solo hecho de escoger una perspectiva afecta ya a la interpretación de la imagen.

En algunos países existen catálogos de recomendaciones elaborados y propuestos por grupos de trabajo como por ejemplo el SWGIT (*Scientific Working Group on Imaging Technology* – Grupo de Trabajo Científico para Tecnologías de Imagen), integrado por fotógrafos, informáticos, investigadores de agencias estatales y federales y representantes de la Universidad y la comunidad investigadora, cuyo propósito consiste en facilitar la integración de las tecnologías de proceso digital de imagen en el sistema legal norteamericano. A tal efecto se dedican a emitir definiciones y recomendaciones relativas a la captura, el almacenamiento, análisis, transmisión telemática e impresión de aquellas imágenes que hayan de ser presentadas como prueba ante los tribunales.

Las recomendaciones SWGIT se extienden a todas las etapas del uso de la imagen digital como prueba. Su planteamiento no difiere en lo esencial de los requisitos exigidos a la prueba electrónica en las investigaciones forenses

convencionales, que ponen énfasis en la necesidad de conservar la imagen tradicional y trabajar sobre copias, así como de llevar un registro documental detallado y exacto de todos los pasos seguidos durante la operación y la exigencia de que todo resultado final en el proceso de análisis de la imagen sea repetible si se sigue el mismo procedimiento que aparece documentado en el informe. Dichos pasos deben ser tan claros que cualquier persona con un conocimiento adecuado del tema pueda entender sin dificultad cómo se ha realizado el análisis y llevarlo a cabo ella misma en el momento preciso. Así mismo las recomendaciones SWGIT establecen la necesidad de aplicar métodos claros y no distorsionantes de la evidencia a la hora de aplicar ajustes que en algún momento pudieran resultar necesarios para procesar la imagen digital, como mejoras de luminosidad o contraste, restauración, aplicación de *plugins* para aumentar la nitidez de la imagen o eliminar ruido. Lo mismo se ha de aplicar en lo relativo a compresión de archivos.

No hace falta decir que todos estos pasos deben llevarse a cabo minimizando la intervención y por razones estrictamente informativas, jamás de estética, de mejora de la composición o para introducir sesgos de ningún tipo. Según SWGIT, para que una imagen sea aceptada como prueba en un tribunal debe cumplir las características siguientes:

- **Fiabilidad:** las imágenes deben ser creíbles, fidedignas, nítidas y sin errores.
- **Carácter reproducible:** los procesos aplicados durante el análisis forense deberán poder repetirse en cualquier momento por otro experto, el cual al hacerlo habrá de obtener los mismos resultados.
- **Cadena de custodia:** debe poder garantizarse que la imagen original ha permanecido en un lugar seguro y únicamente ha tenido acceso a aquella personal autorizado.

### 8.3.3 Buenas prácticas

Todo catálogo de buenas prácticas relativo al tratamiento de imágenes digitales destinadas a servir como prueba en un tribunal ha de constar de un número de puntos y recomendaciones esenciales:

- Archivado de la imagen original.
- Se trabaja únicamente con copias del archivo original.

- Solamente se utilizarán procedimientos válidos para el análisis de la imagen.
- Todos los procesos deberán ser repetibles y verificables.

El archivado de la imagen original, al constituir el primer eslabón en la cadena de custodia, es decisivo para todo lo que viene detrás. La imagen tiene que conservarse en su formato original y permanecer almacenada para fines de prueba. La ubicación del archivo –disco duro, servidor, DVD o cualquier otro medio– no es en sí tan importante. Lo que realmente interesa es que la imagen se mantenga íntegra y esté protegida contra daños o alteraciones de cualquier tipo. El resto de los requisitos se consiguen a través de un método de trabajo ordenado y una documentación impecable conforme a los puntos expuestos en el capítulo 2 de este mismo libro al tratar de los requerimientos de una investigación forense y la intervención del perito ante el tribunal.

Un catálogo de buenas prácticas no garantiza por sí mismo el éxito del caso, pero aportando transparencia y métodos comprensibles reduce la posibilidad de un rechazo de la evidencia. Lo confuso resulta fácilmente atacable. Así mismo el investigador debe ser consciente de que la mayor parte de jueces y fiscales suelen ser inexpertos en tecnologías de la información, cuánto más en una especialidad exótica y de reciente desarrollo como la investigación forense de imágenes digitales. Una operativa impecable y buenos informes ahoran trabajo y sirven a los intereses de su causa legal mejor que cualquier alarde de maestría técnica y autoridad en la materia.

### 8.3.4 Adquisición de imágenes en formato RAW

A la hora de cumplir el requisito de archivado de la imagen original se plantean diversos problemas. Existen pocas cámaras capaces de calcular el *hash* de la imagen después de haber disparado. ¿Resulta asumible la perspectiva de descargar las fotos inmediatamente o lo antes que se pueda a un ordenador portátil, hacer una suma de verificación MD5 o SHA1 y después volver a copiar los archivos JPEG al CD-ROM que finalmente habrá de ser presentado como prueba ante el tribunal, todo ello en el lugar? Si las buenas prácticas significan claridad y métodos directos lo ideal sería que la tarjeta de memoria, una vez obtenidas las fotografías, pasara directamente de la cámara digital a la bolsita precintada de las pruebas. Sin embargo, ¿cómo se puede estar seguro de que el archivo original no es alterado en ningún momento?

La tecnología solo ofrece una respuesta: los datos “crudos” en formato RAW, es decir, una ristra de unos y ceros procedentes de la conversión AC-DC de

las señales eléctricas generadas por el CCD durante la toma fotográfica, no procesados aún por el software de la cámara y recogidos en forma de archivo informático. La información correspondiente a cualquier proceso posterior (mejoras de color, optimización del contraste, eliminación de ruido, correcciones de desenfoque) no forma parte del archivo RAW, sino que se guarda en bloques aparte –archivos sidecar o copias del original en JPG–, con lo cual una imagen RAW resulta lo más parecido a un negativo fotográfico convencional. En caso de duda se podrá desandar el camino y volver sobre la prueba original.

Existen sin embargo algunos inconvenientes, como por ejemplo el gran tamaño de los archivos RAW (hasta diez veces mayor que el de una imagen JPG) y un mayor tiempo para guardarlos en el soporte de datos, lo cual hace más lenta la obtención de la fotografía. Esto puede que ya no sea un problema gracias al avance de la tecnología y a las enormes capacidades de almacenamiento de la actualidad, en un momento en que ya están saliendo al mercado tarjetas de hasta 64 GB. Por el contrario sí que puede resultar un inconveniente la gran variedad de formatos RAW existentes en el mercado. La mayor parte de ellos son propietarios, no existe documentación ni especificaciones para desarrolladores de software y en la práctica cada marca de cámara digital tiene el suyo, adaptado a las características técnicas del CCD y otros componentes electrónicos que emplea. Por si fuera poco los programas de retoque fotográfico no incluyen soporte más que para unos cuantos modelos de cámara de amplia difusión. La aparición reciente de formatos RAW representa un avance considerable en este sentido. Gracias a los archivos DNG de Adobe, por ejemplo, las imágenes RAW obtenidas con cualquier cámara seguirían estando disponibles para cualquier investigación aunque no se disponga del software de descarga de la cámara, o incluso aunque ese modelo de aparato haya dejado de fabricarse.

Un conocimiento adecuado de los formatos RAW permite mejorar la calidad del trabajo de quienes utilizan medios tecnológicos de apoyo para actividades de documentación: peritos judiciales, agentes de policía, periodistas, científicos e investigadores de todo tipo. En estos ámbitos el principal reto consiste no tanto en seguir el ritmo del progreso como en aplicar el respectivo catálogo de buenas prácticas y ser capaces de mantener trayectorias consecuentes y verificables en la búsqueda de la verdad.

## 8.4 METADATOS GRÁFICOS

En el capítulo 4 se habló de los metadatos Exif al tratar sobre la investigación de los diferentes tipos de archivos gráficos. También tratamos de una herramienta muy útil para la extracción de información oculta en archivos JPEG,

PNG y de otros formatos. El análisis forense de imágenes digitales incluye el estudio de las etiquetas de texto que los acompañan en busca de posibles elementos de evidencia.

#### 8.4.1 Exif

El estándar Exif (se escribe con minúsculas, no EXIF) fue desarrollado por la organización sectorial japonesa JEIDA (*Japan Electronic Industry Development Association*) con el propósito de incluir metadatos en determinados formatos de archivos de imagen, principalmente JPEG, TIFF, WAV, AVI, etc. Al hacer una fotografía con una cámara digital y trasladarla al ordenador, la primera impresión es que los únicos datos disponibles acerca de la imagen son los que añade el sistema operativo: nombre –asignado automáticamente por la cámara o por el usuario según sus preferencias–, fecha de creación, acceso, tamaño, etc. Si la cámara soporta la especificación Exif, la imagen lleva en su interior una cantidad apreciable de datos: fecha de realización de la foto (independiente de la fecha y hora manejadas por el sistema de archivos), apertura del diafragma, velocidad de disparo, exposición, distancia focal del *zoom*, sensibilidad ISO, empleo del *flash*, etc. Así mismo incluye información sobre el fabricante y su producto: marca, modelo, motor de software empleado para procesar los datos del sensor, etc. En los dispositivos más avanzados Exif también soporta coordenadas geográficas introducidas por el GPS e incluso información sobre *copyright* y derechos de autor, aunque esto último solo en cámaras profesionales de gama alta.

En Windows XP, Vista y 7 podemos acceder a parte de esta información haciendo clic con el botón derecho del ratón sobre un archivo de imagen y seleccionando **Propiedades**. Este método no es recomendable en una investigación forense porque existe el riesgo de alterar el archivo. En sistemas Mac OS X la misma información puede recuperarse mediante Finder expandiendo la sección correspondiente. Con Linux los metadatos Exif se visualizan a través de la pestaña información o con un visor de gráficos.

#### 8.4.2 IPTC-IIM y Adobe XMP

Exif no es el único sistema de información que existe para etiquetar imágenes digitales. IPTC-IIM (*IPTC Information Interchange Mode*), cuyos orígenes se remontan a finales de los años 70, fue creado por un consorcio que agrupa a las agencias de noticias y empresas de comunicación más importantes del

mundo. Estas compañías estaban interesadas en desarrollar un estándar para el intercambio de informaciones gráficas destinadas a la industria editorial y los medios de comunicación.

Otro sistema importante es la plataforma de metadatos ampliable Adobe XMP. Se trata de una tecnología basada en estándares abiertos para captura e intercambio de metadatos en medios digitales y flujos de trabajo relacionados con el proceso digital de imágenes. XMP utiliza XML para describir los metadatos. Esto supone algunas ventajas: compatibilidad con aplicaciones y desarrollos, conservación de los metadatos a lo largo del flujo de trabajo, trazabilidad de procesos y la seguridad de que las inversiones realizadas en equipos y software no van a quedarse obsoletas de un año para otro.

#### 8.4.3 Instalación y manejo de Exiftool

Las herramientas de información del sistema operativo y los programas de retoque fotográfico no muestran el conjunto total de etiquetas, sino solo aquellas que consideran relevantes para sus fines, normalmente gestionar archivos o facilitar su búsqueda en grandes repositorios de imágenes. Además al abrir un archivo puede quedar alterada la cabecera del mismo, quedando de este modo comprometida la evidencia del investigador. Por este motivo interesa disponer de una utilidad con la cual se puedan analizar los archivos sin tener que abrirlos. En el capítulo 4 se habló sobre Exiftool, un software recomendable por varias razones. Para empezar por su facilidad de uso y la posibilidad de redireccionar la salida del comando creando así archivos de texto con los resultados del análisis, que después pueden ser examinados *off line* y agregados al informe de investigación. Exiftool ofrece otras ventajas: se trata de un desarrollo de código libre capaz de recuperar bastante más información que cualquier otro software, incluyendo metadatos de archivos de otros formatos como por ejemplo ejecutables, multimedia (AVI, MPEG, WAV, FLAC, MP3, Flash, Quicktime, Kodak, Ricoh, APE, Vorbis), documentos Microsoft Office o Adobe PDF, archivos comprimidos ZIP, etc. También permite la búsqueda selectiva y la extracción de datos mediante sentencias condicionales, así como la posibilidad de utilizar tablas de caracteres extraeuropeos (árabe, cirílico, etc.).

En las líneas que siguen se hace referencia a ejemplos de instalación y manejo con Linux, pero también existen versiones para Windows. Si la distribución de Linux que el investigador utiliza está basada en Ubuntu podrá encontrar Exiftool en cualquiera de los repositorios de software de este sistema operativo:

```
sudo apt-get install exiftool
```

Si se quiere instalar Exiftool sobre otra distribución Linux o disponer de la última versión del software el procedimiento resulta algo más complicado. En primer lugar hay que instalar los paquetes de Perl. Por ejemplo, si somos usuarios de Fedora:

```
yum install -y perl-ExtUtils-MakeMaker
```

A continuación hay que descargar Exiftool de la página del desarrollador:

```
wget http://www.sno.phy.queensu.ca/~phil/exiftool/Image-ExifTool-X.XX.tar.gz
```

Las letras “XX” representan al número de versión correspondiente. El resto no es más que rutina:

```
gunzip < Image-ExifTool-X.XX.tar.gz | tar xvf
cd Image-ExifTool-X.XX
perl Makefile.PL
make test
make install
```

#### 8.4.4 Ejemplo de aplicación

Nada ilustrará mejor la sencillez de manejo y la potencia de Exiftool que un ejemplo de uso. Observe estas tres fotografías. La primera es la imagen original sin retocar, tal y como fue descargada desde la cámara digital. Las otras han sido sometidas a una corrección de perspectiva con dos programas bastante populares de retoque fotográfico: Adobe Photoshop Elements y GIMP. Tras haberlas situado en una carpeta de trabajo abrimos una consola de texto y lanzamos contra ellas nuestra herramienta Exiftool redirigiendo la salida del comando a un archivo de texto:

```
exiftool * > resultados.txt
```

Lo que se muestra a continuación no es más que un grupo reducido de todas las etiquetas listadas por Exiftool. El *output* completo del comando comprende media docena de páginas de este libro. Prácticamente la totalidad de las etiquetas contenidas por el archivo son rescatadas y mostradas en forma de tabla por Exiftool en la salida estándar del comando. Entre otras informaciones Exiftool permite conocer por ejemplo la marca, el modelo, el número de serie, la fecha de la fotografía y los ajustes de la cámara.

File	Modification	:	2011:11:26
Date/Time		:	11:23:56+01:00
Make		:	Panasonic
Camera Model Name		:	DMC-LX3
Exposure Time		:	1/400
ISO		:	80
Internal Serial Number		:	(P34) 2009:03:04 no. 0073
Exif Image Width		:	3776
Exif Image Height		:	2520
Focal Length In 35mm		:	24 mm
Format		:	JPEG (old-style)
Compression		:	10752
Thumbnail Offset		:	7284
Thumbnail Length		:	2.8
Aperture		:	1/400
Shutter Speed		:	(Binary data 7284 bytes, use -b option to extract)
Thumbnail Image		:	

Datos Exif



Imagen sin retocar



Corrección de perspectiva con GIMP...



...y con Photoshop Adobe Elements

Figura 8.8. Metadatos Exif

Fijémonos en la etiqueta denominada “Thumbnail”. Los *thumbnails* son imágenes reducidas de la fotografía que se elaboran automáticamente con el objeto de mostrarlas en el visor de la cámara, en la carpeta de trabajo de un ordenador o en la hoja de contactos de un visor de gráficos. Frecuentemente se les encuentra incrustados en el interior del archivo mostrando la imagen original antes de retocarla. La resolución de estos *thumbnails*, con un tamaño tan reducido, no es muy alta, pero en ocasiones pueden llegar a proporcionar la prueba evidente de un intento de manipulación.

La imagen original de la figura 8.8 no convencía al fotógrafo, ya que al haber sido obtenida con un gran angular presenta una distorsión de perspectiva que deforma la planta del edificio. El autor intentó corregirla utilizando para ello los programas de retoque anteriormente mencionados. Sabemos cuáles son porque nos lo dicen las etiquetas Exif:

Software	: GIMP 2.6.10
XMP Toolkit	: XMP toolkit 2.8.2-33, framework 1.5
Creator Tool	: Adobe Photoshop Elements for Windows, version 2.0

#### 8.4.5 Limitaciones

La potencia de Exif no debe hacer olvidar las restricciones impuestas por el entorno legal. Este software no fue diseñado con fines forenses sino para finalidades de gestión y desarrollo. Con él no solo se pueden visualizar etiquetas, sino también modificarlas. Suprimir, alterar y falsear metadatos gráficos resulta tan sencillo que una argumentación basada exclusivamente en etiquetas Exif o XMP no llegaría lejos ante un tribunal. El abogado de la parte contraria la desmontaría en poco tiempo argumentando con razón que desde Photoshop o cualquier otro software de retoque se puede conseguir que los metadatos no cuenten la verdad, sino una historia hecha a medida de quien la narra.

Por consiguiente el investigador no debe hacer en ningún momento afirmaciones caprichosas, sino atenerse estrictamente a hechos objetivos, comprobables y desprovistos de cualquier connotación interpretativa. Evite hacerse un alto concepto de la importancia jurídica de sus descubrimientos, que solo tienen valor en el contexto general de la investigación y referidos a un conjunto de elementos de evidencia digital más amplio.

## Capítulo 9

# HELIX

Existen varias razones para incluir en este libro un capítulo dedicado a Helix. En primer lugar incluye en un solo CD la mayor parte de las herramientas de código libre utilizadas en los capítulos anteriores, con una amplia documentación e instrucciones detalladas de manejo. Por otra parte el diseño dual de Helix permite una gran variedad de usos y configuraciones. El investigador lo puede utilizar para el análisis en vivo de un sistema Windows, simplemente introduciéndolo en la unidad lectora DVD/CD-ROM y esperando a que arranque automáticamente. En las últimas versiones también podrá hacer lo mismo con sistemas Linux y Apple OSX. Además Helix puede iniciarse como un Live-CD en cualquier PC con arquitectura X86 –incluyendo en su última versión Pro las de 64 bits– para desde allí llevar a cabo operaciones de previsualización y búsqueda de evidencia, adquisiciones *post mortem* o tareas de reparación y recuperación de datos. Helix está respaldado por los más destacados especialistas en Informática Forense, algunos de ellos autores de las herramientas a las cuales se ha hecho mención a lo largo de esta obra. Sus manuales de instrucciones, escritos en un inglés sencillo y asimilable sin dificultad, son completos, de lectura amena y abundantes en ejemplos prácticos.

La utilización de una herramienta como Helix aporta al investigador otro beneficio: la oportunidad de recapitular todas las etapas características de una investigación forense –recuperación de informaciones volátiles, adquisición de memoria RAM, elaboración de *hashes* de los soportes de datos, obtención de copias a bajo nivel y análisis forense de los elementos de evidencia recolectados– con todas las herramientas cuyo funcionamiento se ha explicado en las páginas anteriores, todo ello reunido en un solo CD.

Todas las informaciones a las que se hace referencia y las imágenes que figuran en este capítulo son propiedad de e-fense, desarrolladora de Helix (<http://efense.com>), y figuran expuestas en el presente capítulo con la autorización explícita de la empresa.

### 9.1 UNA DISTRIBUCIÓN DUAL

#### 9.1.1 ¿Qué es Helix?

Helix comenzó a desarrollarse en el año 2003 tras haber adaptado para fines forenses una distribución estándar llamada Knoppix, creada como plataforma para reparación de sistemas y aprendizaje de Linux por el informático alemán Klaus Knopper. Knoppix funcionaba –y sigue funcionando– como un Live-CD autoarrancable de Linux. Esto quiere decir que no hace falta instalar el sistema operativo en el ordenador. El usuario arranca desde un CD o desde una llave USB y una vez iniciado el sistema realiza sus tareas, ejecuta programas, navega por Internet y lleva a cabo otras actividades sin tocar el disco duro del ordenador –a no ser que se acceda a él de modo explícito para llevar a cabo cometidos de reparación o rescate de datos–. Finalmente detiene el sistema, extrae el CD-ROM de la bandeja y apaga el ordenador dejándolo en el mismo estado en que lo encontró. Knoppix incluye la posibilidad de ser instalado en el disco duro lo mismo que cualquier otra distribución de Linux. La verdadera utilidad de un live-CD, sin embargo, no consiste en eso, sino en permitir el empleo ocasional de un ordenador sin tener que realizar modificaciones de ningún tipo en el sistema operativo ni en sus particiones. Esta capacidad, junto con el hecho de que los *scripts* de arranque hayan sido modificados para que el montaje de todas las particiones se lleve a cabo exclusivamente en modo de solo lectura, convierten a Helix en una herramienta ideal para la adquisición y el análisis *post mortem* de plataformas Intel 86.

#### 9.1.2 Características y novedades

Las últimas versiones, como por ejemplo Helix3 Pro™, liberada en 2009, suponen un replanteamiento del proyecto original basado en Knoppix. Para empezar, la plataforma autoarrancable se traslada a una distribución Ubuntu modificada. La parte que inicia en vivo –es decir, tras haber introducido el CD en la unidad de lectura/grabación de un ordenador en funcionamiento–, y que antes solo funcionaba bajo Windows 9X, 2000 y XP, ahora sirve para realizar adquisiciones de información volátil (hora, fecha, usuarios conectados, estado de las conexiones, memoria, etc.) no solo en todas las versiones del sistema de Microsoft incluyendo Vista y 7, sino también en sistemas Linux y Apple OSX.

Parte del código Ubuntu subyacente ha sido modificado para optimizar la ejecución de tareas forenses. Esto incluye la recompilación de herramientas y librerías con el fin de que el sistema atienda al cumplimiento de requisitos imprescindibles para la adquisición y el análisis forense de soportes de datos. Entre otras características de interés para el investigador, Helix, en su última versión denominada Pro™, presenta las que se mencionan a continuación:

- Helix jamás utiliza las particiones de intercambio halladas en un sistema.
- En el arranque de Helix el montaje automático de particiones establece el acceso a las mismas en modo de solo lectura.
- Helix posibilita el acceso a todos los sistemas de archivos identificados por Linux, como por ejemplo ext2, ext3, vfat, NTFS, ReiserFS, xfs, jfs y muchos otros.
- Helix incorpora gran cantidad de herramientas forenses para el estudio *in vivo* y *post mortem* de sistemas operativos, la mayor parte de código libre. Entre ellas está incluida la práctica totalidad de las que el investigador ha tenido ocasión de conocer en los capítulos anteriores.
- Helix incluye una aplicación ejecutable para sistemas Windows, Linux y Mac que permite extraer de ellos la información volátil durante la intervención inicial del investigador.

Cuando al principio decíamos que Helix es una distribución dual nos referímos a que posee dos modos de funcionamiento: uno *en vivo* (no confundir con el modo autoarrancable característico de las distribuciones Live CD) al que se accede simplemente insertando el CD de Helix en la unidad lectora de un ordenador con un sistema operativo –Windows, Linux o Apple OSX– y ejecutando las herramientas forenses sobre el entorno que se quiere investigar para de este modo extraer su información volátil y otros elementos de evidencia; y un segundo modo como CD autoarrancable, que permite iniciar un ordenador apagado con el objeto de acceder a su disco duro para examinar sus contenidos o realizar una imagen a bajo nivel por medio de las herramientas de adquisición habituales (dd, Adepto, FTK Imager, etc.).

### 9.1.3 Obtención de Helix

La versión 3, ya basada en Ubuntu y con el entorno GNOME, de Helix puede descargarse en forma de imagen ISO desde <http://e-fense.com>. Si el lector es

fan de Knoppix y quiere experimentar con las versiones anteriores de Helix, tendrá que buscarlas en diversos sitios de Internet y en los FTP públicos de universidades y centros académicos. Por el contrario Helix Pro™, con su nuevo diseño y capacidad para acceder a otros sistemas operativos, solo está disponible como prestación de un sistema de suscripciones de e-fense. La suscripción, que requiere el pago de una cuota anual, permite la descarga de Helix3 Pro™ y sus actualizaciones así como la participación en foros y un servicio de soporte técnico. También se proporciona al suscriptor acceso a recursos adicionales como seminarios *on line*, artículos especializados y diversas informaciones periódicas sobre análisis forense digital.

Una vez obtenida la imagen ISO debe comprobarse su integridad (figura 9.1) extrayendo un *hash* y comparándolo con el que figura en la página de descargas. Finalmente basta volcar la imagen sobre un CD en cualquier programa de grabación como Nero Burning ROM, Roxio CD Creator o k3b (bajo Linux) y con ello el investigador ya tiene listo su Live-CD Helix listo para utilizar. Esto lo puede hacer de dos maneras que se describen a continuación: *en vivo* con un sistema en funcionamiento o arrancando el ordenador con Helix después de haberlo reiniciado.

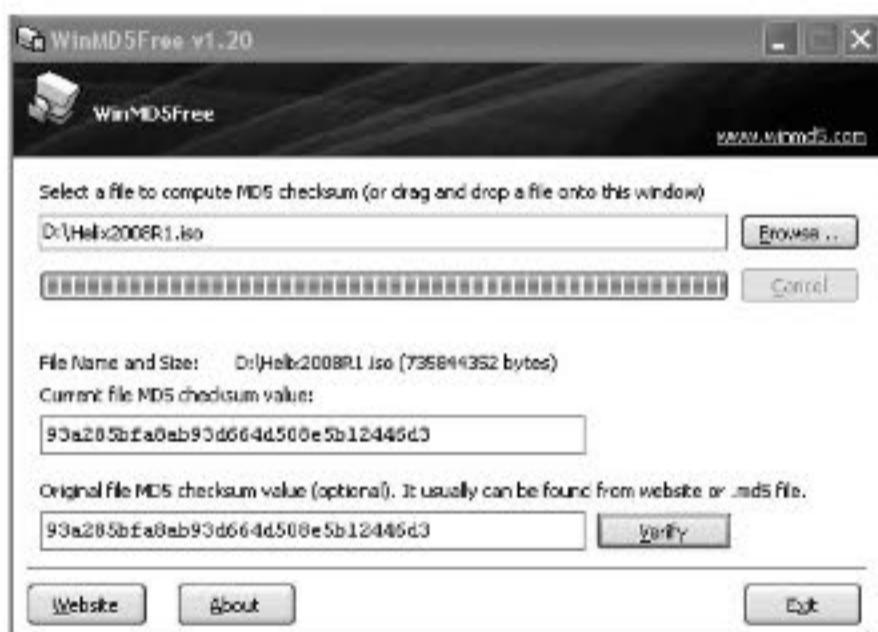


Figura 9.1. No olvide calcular la suma de verificación de Helix

### 9.1.4 Arranque *en vivo*

El investigador al que se le ha encomendado una intervención inicial (*first response*) en un entorno Windows, Mac o Apple OSX, inserta su CD con Helix3 Pro™ y ejecuta las aplicaciones correspondientes desde la carpeta con el ícono del CD/DVD correspondiente a la unidad lectora. Si el sistema es de tipo MS-Windows con *autorun* activado el interfaz de usuario de Helix se inicia automáticamente. En sistemas Linux y Apple OSX por el contrario habrá que

ejecutar la aplicación correspondiente haciendo clic sobre el archivo ejecutable. Los binarios que componen las herramientas a bordo de Helix están compilados estáticamente. Esto quiere decir que durante su funcionamiento no necesitan recurrir a las librerías del sistema, lo cual ayuda a reducir el impacto sobre el entorno investigado al mismo tiempo que se evita la posibilidad de un funcionamiento defectuoso o poco fiable en caso de haber instalado un troyano o un *rootkit* en el sistema.

Es importante recordar que no todos los binarios de Helix están compilados estáticamente, sino solamente los de aquellas herramientas que resultan de vital importancia para la intervención del investigador y el rescate de la información volátil. De hecho el entorno gráfico de Helix, por exigencias de espacio, accede a gran número de librerías de los sistemas subyacentes para poder ejecutarse. Esto es algo que hay que tener en cuenta y comprender a la hora de realizar los trabajos de análisis posteriores y finalmente el informe forense. Al efecto de explicar mejor el funcionamiento de Helix y la forma en que afecta a las librerías de los entornos sobre los cuales se ejecuta en vivo, e-fence proporciona, tanto en su sitio web como en la documentación de Helix, un listado completo de librerías y utilidades de los diferentes sistemas operativos a los que Helix accede durante su ejecución en vivo.

A diferencia de sus predecesoras, que únicamente funcionaban sobre Windows XP y versiones anteriores del sistema de Microsoft, la versión actual, denominada Helix3 Pro™, es capaz de iniciarse sobre una variedad de sistemas operativos. Ha sido probada sobre Windows 2000, Windows XP, Windows 2003 Server, Vista, Windows 2008 Server y Windows 7, y también sobre Ubuntu y Apple OSX 10.4, cubriendo con ello una vasta mayoría de las plataformas utilizadas actualmente y con las que un investigador forense tendría la oportunidad de encontrarse durante su trabajo.

### 9.1.5 CD autoarrancable

Helix arranca desde cualquier ordenador con arquitectura x86, que de la misma manera es como decir la mayor parte de los ordenadores existentes en el mercado. Lo mismo que Ubuntu y otras distribuciones Live-CD, forma al iniciarse un entorno autocontenido en Linux, al cual se han hecho algunos ajustes para adaptarlo a una utilidad forense, con las herramientas dispuestas en el interfaz de usuario y los menús característicos de Ubuntu. Helix no es la primera distribución de estas características con herramientas diseñadas para realizar tareas de tipo forense. Sin embargo, mientras otros Live-CD de Linux como Knoppix-STD, Backtrack o SystemRescueCD incluían software para cometidos de seguridad informática y reparación de sistemas, Helix se concentra exclusivamente en la respuesta ante incidentes y el análisis forense.

## 9.2 HELIX SOBRE UN SISTEMA EN FUNCIONAMIENTO

Cuando se accede a un sistema en funcionamiento, independientemente de que el investigador lo haga con Helix o con cualquiera de las herramientas de las que se habló en los capítulos 4 y 5 al tratar sobre la recuperación de las informaciones volátiles, se producen cambios en el ordenador sospechoso. Por este motivo la utilización de Helix en su modalidad de funcionamiento en vivo es una medida a la que solamente se debe recurrir en circunstancias especiales, por ejemplo cuando por alguna razón el sistema no pueda ser desconectado o interese conocer la información volátil tras un incidente de seguridad y antes de que el ordenador sea apagado para realizar la adquisición forense de sus soportes de datos.

En el manual de instrucciones y la documentación de Helix se advierte que utilizar esta herramienta introduce modificaciones en el sistema investigado, con el riesgo que ello supone para la integridad de los elementos de evidencia. Recuerde que el solo hecho de insertar y extraer el CD ya produce en un entorno informático cambios que posteriormente podrían hacer que la prueba forense quede invalidada, a no ser que se alegue una razón de peso para haber utilizado herramientas de adquisición en vivo. Esta advertencia ha de ser tenida en mayor consideración por cuanto la modalidad de funcionamiento en vivo de Helix requiere privilegios de administrador en el ordenador intervenido. Nunca se insistirá lo bastante en la necesidad de reflexionar, antes de hacer uso de una herramienta de este tipo, si su utilización va a aportar más elementos de evidencia de los que puede destruir. Todo depende de los objetivos de la investigación y las circunstancias específicas del caso.



Figura 9.2. Interfaz de Helix

## 9.2.1 Interfaz

Introduzca el CD en la unidad de lectura. Si *autorun* está habilitado (lo cual suele ser frecuente en plataformas Windows), la aplicación Helix (figura 9.2) arrancará automáticamente. En caso contrario deberá hacer doble clic sobre el ícono de la aplicación en la carpeta del CD tras haberse desplazado hasta ella con el Windows Explorer. Mac OSX y Linux carecen de funcionalidades *autorun*, por lo que deberá buscar la aplicación y ejecutarla.

El interfaz de usuario de Helix, cuando se activa sobre un sistema en funcionamiento, es autoexplicativo. Un sistema de pantallas guía al usuario a través de los diversos menús de herramientas. En la versión más reciente ha habido cambios sustanciales con respecto a las anteriores. Hasta Helix 3 los iconos representativos de las diferentes categorías funcionales estaban situados en la parte izquierda de la ventana. Pulsando sobre cualquiera de ellos se accedía a diversos grupos de herramientas situados a la derecha en forma de un montaje similar a una presentación de diapositivas. Así, por ejemplo, seleccionando la figura de un ordenador se podía obtener información general del entorno, como la versión del sistema operativo, el estado de las conexiones de red, nombre y datos del propietario, procesos en ejecución, etc. Por debajo del ícono anterior, una cámara fotográfica en miniatura permitía al investigador realizar adquisiciones forenses de diversos elementos del sistema: discos duros, disquetes o memoria RAM, guardando sus imágenes en una unidad de almacenamiento externa. Y así sucesivamente.

Por el contrario el diseño de la versión más reciente, Helix3 Pro<sup>TM</sup>, es más profesional y está más adaptado a las necesidades de un investigador profesional. Obviamente los desarrolladores se han propuesto mejorar el software aproximándolo a los estándares de estilo y aspecto visual típicos de las aplicaciones comerciales. La pantalla principal muestra en la parte superior izquierda una barra de tareas compuesta por una hilera de iconos correspondientes a las principales áreas de actividad del investigador: información del sistema, adquisición, obtención de *hashes* de los diferentes soportes –algo imprescindible para asegurar el primer eslabón de la cadena de custodia– y búsqueda de imágenes en diversos formatos gráficos. Los ejemplos siguientes están referidos a Helix3 Pro<sup>TM</sup>.



Figura 9.3. Información del sistema

## 9.2.2 Información del sistema

Las pantallas de información proporcionan los elementos básicos con los que el investigador habrá de comenzar su informe: administrador del sistema, dirección IP del *host*, dirección MAC de su adaptador de red, la máscara de subred utilizada en la LAN, el nombre de los usuarios conectados y toda la información relevante del sistema operativo (figura 9.3).



Figura 9.4. Información volátil

### 9.2.3 Examen de la información volátil

Siguiendo con el esquema que ya se ha visto en los capítulos correspondientes al análisis forense de sistemas Windows y Linux, el investigador, con solo bajar un peldaño en el panel de mandos, podrá llevar a cabo de manera sistemática un examen de los datos volátiles como la información de red, procesos y servicios en ejecución, variables de entorno, tablas de enrutamiento y ARP, aplicaciones instaladas y otros elementos de evidencia (figura 9.4). Por el momento esta utilidad funciona únicamente en Windows, pero en versiones posteriores de Helix se tiene la intención de extenderla a entornos Linux y OSX.

### 9.2.4 Información de discos

Helix presenta la disposición de discos en un esquema jerárquico encabezado por los volúmenes físicos y colgando de ellos los volúmenes lógicos y las particiones. La lista incluye todos aquellos dispositivos accesibles en el sistema sobre el que se está ejecutando Helix: discos duros, unidades DVD/CD-ROM e incluso medios de almacenamiento remotos que estén montados en ese momento. Por ejemplo se podría ver un ordenador Apple haciendo un *backup* o recursos de red en entornos Unix accesibles a través de SAMBA. Para consultar los datos correspondientes a un disco no hay más que pulsar sobre él y se mostrarán sus características: tamaño, fabricante, número de serie y si tiene habilitado algún tipo de encriptación para discos completos. En este último caso el ícono del disco físico llevará un candado encima y la pestaña de información mostrará el tipo de encriptación.



Figura 9.5. Información sobre discos y soportes de datos

Helix reconoce seis sistemas de encriptación diferentes:

- Bit Locker
- TrueCrypt
- SafeBoot
- Luks
- PGP-PGPGuard
- PGP-BootGuard

Seleccionando una unidad lógica se mostrarán en la pestaña de información todas las características esenciales de la misma: sistema de archivos, punto de montaje, sistema de encriptación utilizado, etc.

En la figura 9.6 se puede apreciar por ejemplo una partición cifrada con TrueCrypt.



Figura 9.6. Partición cifrada con TrueCrypt

### 9.2.5 Información de memoria RAM

Finalmente al seleccionar el apartado de memoria (figura 9.7) se obtiene información sobre la cantidad de RAM instalada en el sistema, tanto física como virtual. En el apartado siguiente se explica el procedimiento para adquirir la memoria RAM de un sistema en funcionamiento.

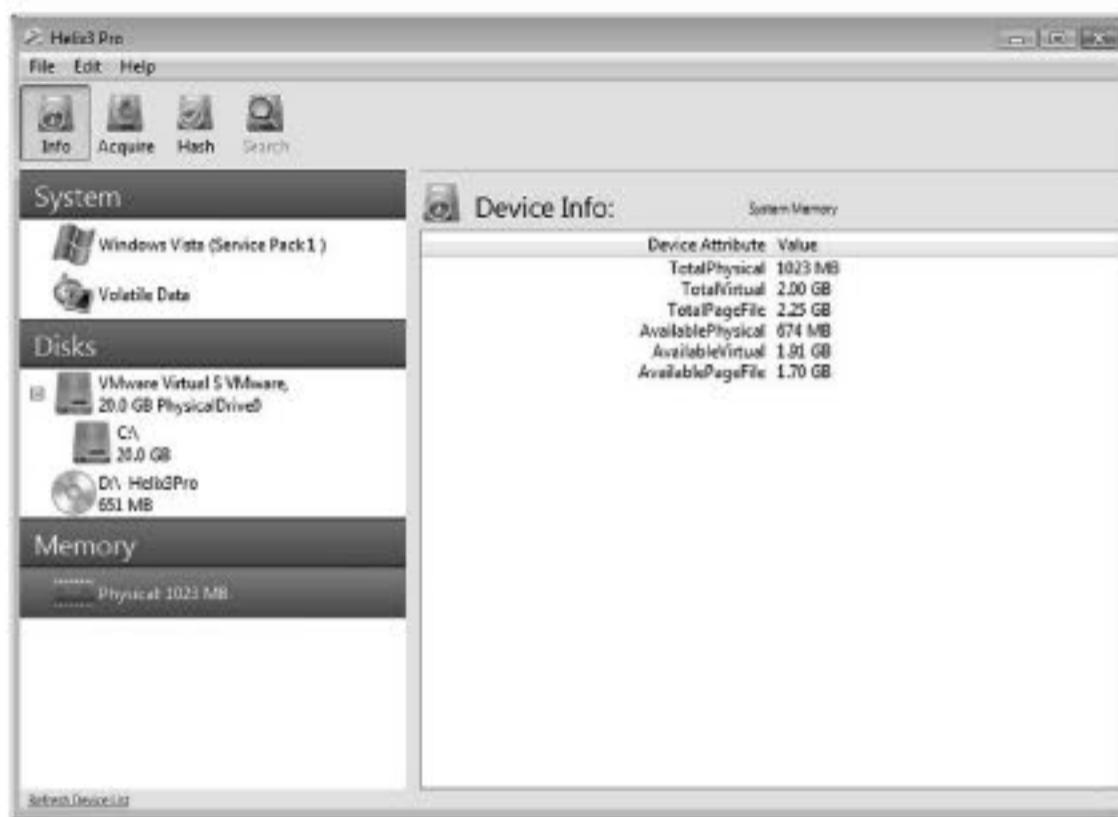


Figura 9.7. Memoria del sistema

## 9.3 ADQUISICIÓN DEL SISTEMA EN VIVO

### 9.3.1 Orden de volatilidad

Para adquirir un sistema en funcionamiento se ha de proceder de manera sistemática siguiendo el orden de volatilidad tal y como se explica en el documento RFC 3227 (<http://www.faqs.org/rfcs/rfc3227.htm>). El orden de volatilidad exige que los datos de un sistema vivo sean recolectados con prioridad inversa a su tiempo de persistencia en el sistema para evitar la pérdida de elementos de evidencia que podrían resultar importantes. Por este motivo la adquisición de la RAM, suponiendo que sea posible, tiene precedencia sobre la de otros datos como las conexiones de red abiertas o los procesos en ejecución. En el último puesto de esta escala de prioridades se hallarían los discos duros y los dispositivos de almacenamiento externos.

En todos los sistemas el orden de volatilidad suele ser el mismo y está compuesto por las etapas que se mencionan a continuación:

1. Memoria RAM y caché.
2. Archivo de paginación y zona de intercambio.
3. Estado de la red y conexiones abiertas (tabla de enrutamiento, caché ARP, etc.).

4. Procesos en ejecución.
5. Información del sistema de archivos.
6. Discos duros.

### 9.3.2 Adquisición de memoria RAM

En otro capítulo de este libro se ha hablado de la adquisición de memoria RAM exponiendo las complicaciones técnicas y jurídicas de los procedimientos altamente intrusivos que a veces son necesarios para llevarla a cabo. Sin embargo, durante los últimos años la captura de memoria en sistemas en funcionamiento se ha convertido en objetivo prioritario de toda investigación forense por su capacidad para aportar elementos de evidencia que de otro modo se perderían irreversiblemente al ser desconectado el sistema para realizar la imagen a bajo nivel de su disco duro. En Helix, gracias a un controlador especial, este cometido se resuelve de manera sencilla para todas las versiones de Windows, incluyendo Vista y 7, tanto de 32 como de 64 bits. Helix también permite adquirir la RAM de un sistema Linux a través del dispositivo /dev/mem. En Apple OSX, sin embargo, todavía no es posible llevar a cabo la adquisición de memoria RAM porque el dispositivo /dev/mem no tolera accesos de ningún tipo, ni siquiera del administrador del sistema, y tampoco existen controladores que permitan manipulaciones de la RAM en espacio de usuario.

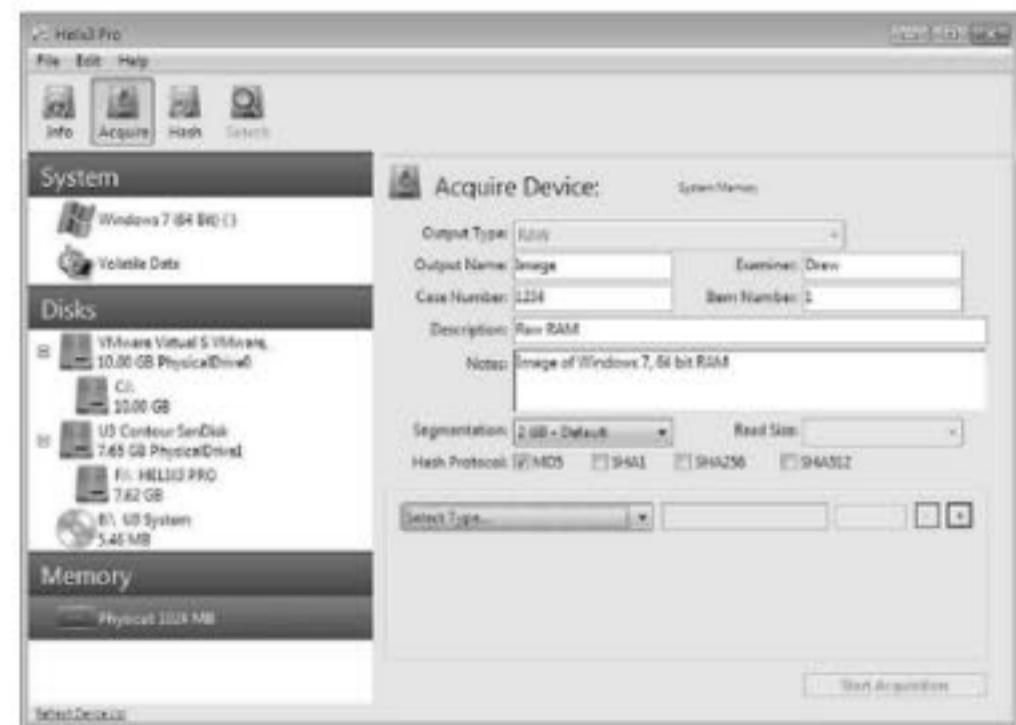


Figura 9.8. Adquisición de memoria RAM

En Helix la adquisición de memoria RAM se realiza desde el apartado Adquirir (Acquire) de la barra superior de herramientas (figura 9.8). Un formulario pedirá al investigador que introduzca datos referentes al caso y los parámetros

deseados para la adquisición, incluyendo la posibilidad de calcular *hashes* y segmentar la imagen en archivos de tamaño conveniente. Cuando todas las opciones hayan sido cumplimentadas deberá seleccionarse una unidad de destino – generalmente una llave USB o un disco duro externo –. También se puede enviar la imagen a otro ordenador de la red local a través de Helix Pro Receiver (ver apartado 9.3.6), de manera parecida a como se hacía con netcat en el capítulo 4.



Figura 9.9. Buscando una ubicación para el archivo de imagen RAM

Para comenzar la adquisición basta pulsar la tecla **Start Acquisition** después de que todos los requisitos anteriores hayan sido cumplimentados. Entonces la imagen de la RAM comenzará a transferirse a un archivo en el dispositivo de almacenamiento o la ubicación de red seleccionada como destino (figura 9.9). Una barra de progreso indicará el estado de la operación. Después de haber finalizado la adquisición, la imagen será verificada automáticamente y el resultado final serán tres archivos en la unidad de destino: la imagen forense de la RAM, un documento que servirá para acreditar la cadena de custodia y un archivo de texto con información relativa a la imagen forense.

### 9.3.3 Recolección de información volátil

Además del volcado de RAM, Helix3 Pro™ ofrece la posibilidad de extraer información existente en memoria sobre determinados aspectos de interés incorporándolos automáticamente a un informe forense en formato PDF o TXT. El documento, al igual que la imagen de memoria RAM, podrá ser guardado localmente en un medio de destino o enviado a través de la red por medio de Helix3 Pro™ Receiver. La recolección automatizada de datos volátiles solamente está soportada en Windows. Los desarrolladores tienen sin embargo la intención de extenderla en el futuro también a OSX y Linux.

La operación es configurable y admite la selección de apartados de información concretos marcándolos convenientemente en un menú cuyo desglose comprende tres grandes apartados:

- Red (tablas ARP y de enrutamiento, interfaces, conexiones abiertas).
- Sistema (controladores, información de volumen, variables de entorno, aplicaciones instaladas).
- Procesos (procesos y servicios en ejecución).

Después de haber seleccionado lo que desea, el investigador no tiene más que elegir el formato de salida y el destino para su informe. Finalmente pulsar **Start Acquisition**. El proceso requiere apenas medio minuto sin necesidad de teclear ningún comando, con la ventaja de que el ahorro de tiempo (apertura de la consola de texto, búsqueda de herramientas en el CD, ejecución de las PsTools, etc.) supone en un sistema recién intervenido, en el que cada segundo cuenta o el más mínimo error de sintaxis del investigador puede resultar decisivo a la hora de preservar o perder elementos de evidencia.

Una vez terminada la adquisición el investigador podrá examinar su informe abriendo el archivo en su ubicación de destino. Para enviar el documento a través de la red local deberá seleccionar la opción correspondiente en el menú de formato de salida (**Text over Network**).



Figura 9.10. Adquisición automatizada de informaciones volátiles

### 9.3.4 Imágenes de discos

En principio podría parecer que la realización de imágenes forenses a bajo nivel tiene más sentido si se hace *post mortem* después de haber reiniciado la máquina o haber extraído el disco duro. No siempre es así. A veces no resulta

posible apartar el ordenador del flujo de producción de la empresa. Y si aquel tiene activado algún sistema de encriptación de disco completo, la imagen forense que obtengamos estará cifrada y no servirá para nada. Por ello es preferible realizar la adquisición con el ordenador en funcionamiento. Helix3 Pro™, en su modalidad de ejecución sobre sistemas vivos, ayuda al investigador a sortear estos inconvenientes, ya que es capaz de detectar los principales mecanismos de encriptación de disco completo y adquiere el soporte a través de los propios controladores que implementan la función de cifrado.

Helix3 Pro™ presenta el soporte de datos en forma de una estructura jerárquica que permite obtener imágenes forenses de discos dinámicos e incluso de un sistema RAID. Basta con seleccionar la unidad lógica y se creará una imagen completa del medio en un archivo similar al que se obtuvo anteriormente al adquirir la memoria RAM (figura 9.11). El investigador podrá elegir entre un formato RAW (dd) o EnCase (E01). Así mismo la imagen de la unidad lógica podrá ser guardada en un medio de almacenamiento local o enviada a través de la red con la misma instancia de Receiver utilizada previamente para el rescate de la información volátil.

Los archivos de imagen suelen tener un tamaño considerable, por lo que deberá asegurarse de que dispone de espacio suficiente tanto en la unidad de destino como en la ubicación remota. La utilidad **Hash** (en la barra superior de herramientas) permite obtener sumas de verificación de los dispositivos adquiridos y salvar a un archivo los valores correspondientes para el uso posterior de los mismos con fines forenses.



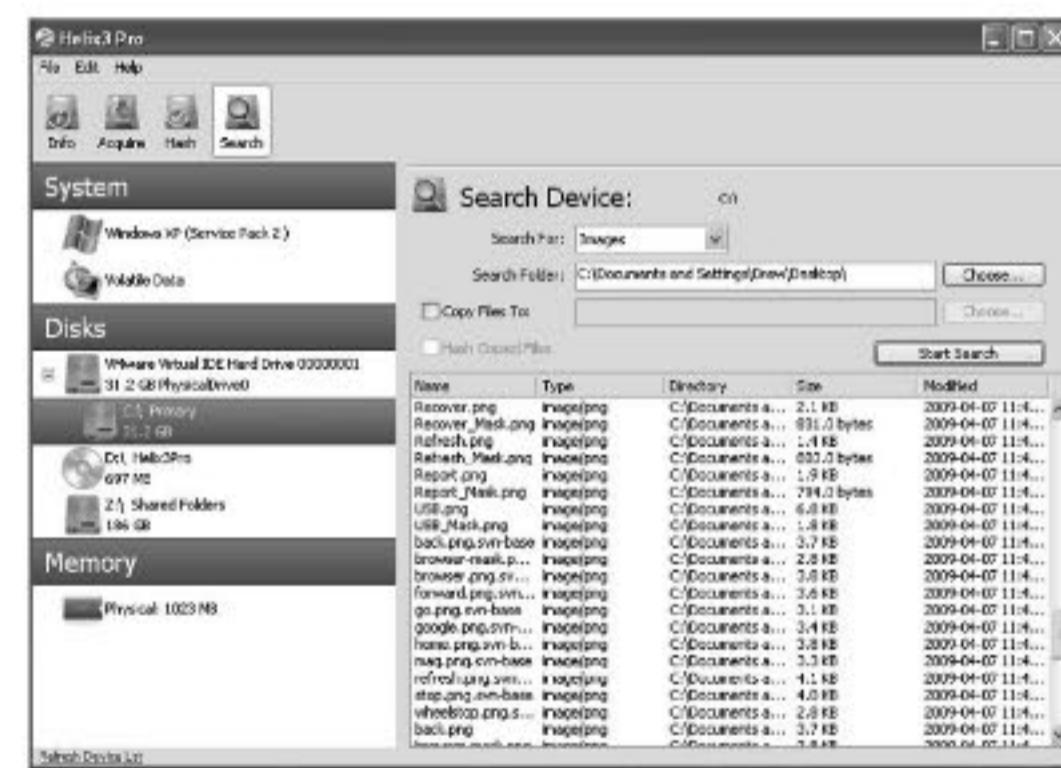
*Figura 9.11. Adquisición de soportes de datos.*

### **9.3.5 Examen de un sistema en funcionamiento**

El botón **Search** en la barra de herramientas permite al investigador llevar a cabo una previsualización del sistema en busca de los tipos de archivo siguientes:

- Documentos.
  - Imágenes.
  - Vídeos.
  - Contenido completo.

Una vez pulsado el botón lo único que hay que hacer es navegar hasta la carpeta correspondiente y visualizar sus contenidos. La búsqueda es recursiva y admite filtrado por tipo de archivo. Los elementos de evidencia localizados pueden guardarse en un medio de destino mediante la opción **Copy Files To:**. Una ventaja adicional de esta utilidad es que Helix permite la extracción de archivos detectándolos al vuelo no por medio de sus extensiones, sino examinando sus encabezados o números mágicos.



*Figura 9.12. Examen del sistema con Search*

### **9.3.6 Helix3 Pro™ Receiver**

Receiver es una utilidad similar a netcat que se ejecuta a través de un interfaz gráfico y permite trasladar a otro ordenador de la red los elementos de evidencia, documentos de informe sobre datos volátiles y archivos de imagen

obtenidos mediante Helix. Receiver funciona en Windows, OSX y Linux. Una de sus características, que lo diferencia de netcat, consiste en la posibilidad de trabajar con más de una imagen a la vez. Receiver permite al investigador establecer una contraseña para hacer más segura la transmisión. Esta contraseña no encripta transmisiones pero hace que las mismas puedan ser recibidas únicamente por otro sistema con Helix3 Pro™ que esté utilizando la misma contraseña.

Una vez elegido el destino para los archivos o las imágenes, Receiver crea de manera automática todos los directorios y subdirectorios que se necesitan. El investigador deberá indicar el puerto que desea utilizar para la comunicación y elegir el tamaño de archivo máximo en caso de que las circunstancias (por ejemplo el uso de un medio de destino formateado con FAT32) le obliguen a dividir el flujo de datos en segmentos de tamaño manejable. Este valor viene ajustado por defecto a 2 GB. Finalmente no queda más que indicar al software que se ponga a la escucha en la red (**Listen For Connections**). Todo esto hay que hacerlo en el medio de destino, o de modo más específico en la aplicación Helix que esté funcionando en el sistema del *host* al cual ha de ser transferida la imagen forense.

En Helix3 Pro™, cuando el usuario selecciona la opción **Image to Helix3 Pro Receiver**, es posible enviar la imagen adquirida a otro sistema en el que se esté ejecutando un programa Receiver. En el ordenador de origen, tras haber pulsado en la barra de herramientas el botón **Adquirir** y una vez introducida la información de acompañamiento (nombre del investigador, número de caso, tamaño de segmentación, notas del investigador y algoritmo para el cálculo del *hash*), se ha de pulsar **Setup** para proseguir. Inmediatamente aparecerá una ventana con diversos campos para configurar el otro extremo de la conexión: dirección IP, puerto, contraseña –caso de haberse configurado una– y nombre del archivo.



Figura 9.13. Receiver

Helix3 Pro™ y Helix3 Pro™ Receiver se sirven de un código especial para detectarse automáticamente en la red. Esto quiere decir que si el investigador ejecuta una instancia de Receiver y a continuación Helix3 Pro™, el mismo software se encarga de definir los parámetros de conexión. Puede haber un problema en caso de que el *firewall* de Windows esté activado. Esta contingencia, sin embargo, se soluciona permitiendo el acceso (figura 9.13). La ventana de configuración de Helix3 Pro™ completa automáticamente la conexión insertando la IP y el puerto en el que Receiver está escuchando. El investigador no tiene más que guardar los parámetros pulsando en el botón **Save**. Entonces la ventana desaparece dejando ver de nuevo el menú principal. Para que la imagen comience a transferirse al ordenador de destino lo único que queda es pulsar **Start Acquisition**.

Una barra de progreso indicará el avance realizado por la tarea y el caudal de datos transferidos. Al otro lado del túnel Receiver mostrará el estado de la conexión entrante y la velocidad de transferencia. La imagen se guarda en una carpeta creada de manera automática por Receiver cuyo nombre tendrá este formato: “AAAA-MM-DD HH.MM.SS – 192.168.X.X” (fecha indicada en año, mes y día; hora en horas, minutos y segundos, y dirección de la red local). En esta carpeta el investigador encontrará tres archivos: la imagen forense, un documento en PDF para levantar acta del establecimiento de la cadena de custodia y un *log* con información relativa al transcurso de la operación y el *hash*.



Figura 9.14. Permitiendo el acceso a través del cortafuegos

Helix3 Pro™ permite al investigador la realización de varias imágenes forenses al mismo tiempo, lo cual resulta de gran utilidad para atender a las necesidades del juzgado y las otras partes litigantes. Con solo pulsar el botón marcado con el signo “+” se pueden añadir al proceso de adquisición forense

nuevos destinos en los que se guardarán las correspondientes copias de la imagen forense y de su información de acompañamiento.



Figura 9.15. Helix LiveCD

## 9.4 HELIX AUTOARRANCABLE

Helix también puede utilizarse como un Live-CD de Linux. Esta modalidad de funcionamiento, como se ha explicado con anterioridad, es diferente a la de la ejecución de Helix sobre un sistema “vivo”. El Live-CD permite arrancar el ordenador desde un medio diferente a aquel en el que se encuentra instalado su sistema operativo –generalmente el disco duro–. Antes se ha dicho que las primeras versiones de Helix estaban basadas en una distribución llamada Knoppix, muy popular entre técnicos de sistemas y estudiantes. Knoppix a su vez había nacido como un desarrollo de Debian, que junto con Red Hat y Slackware fue una de las tres distribuciones históricas que hicieron posible el salto de Linux a un público más amplio desde los reducidos círculos de especialistas en los que había estado confinado durante los primeros años de su desarrollo.

### 9.4.1 Live-CD Linux

Las versiones posteriores a Helix3 están basadas en Ubuntu –a su vez otro desarrollo de Debian–. Helix arranca desde el CD creando un entorno de Linux autocontenido. Para cumplir los requisitos típicos de una investigación forense, sin embargo, ha sido modificado de modo que el sistema pueda detectar el hardware durante el arranque configurando los accesos al mismo en modo estricto de solo lectura. Será imposible acceder a medios de cualquier tipo (discos duros, unidades

externas, etc.) aunque se disponga de permisos de administrador. Si se quiere cambiar algo o habilitar el soporte de destino para llevar a cabo una adquisición forense será necesario desmontar el volumen, abrir una consola de texto y volverlo a montar de manera explícita mediante el comando “mount”. Por ejemplo, si queremos guardar elementos de evidencia en un disco duro externo formateado con una partición FAT32 y asociado al dispositivo sdb, este sería el modo de hacerlo:

```
mkdir directorio_montaje
mount /dev/sdb1 -t vfat ./directorio_montaje
```

Un usuario habitual de Knoppix o incluso Ubuntu –sobre todo si antes ha trabajado con el entorno de ventanas KDE en lugar de GNOME, que es el que viene instalado en Helix– puede sentirse desorientado por la diferente disposición de elementos en los menús, sobre todo a la hora de realizar operaciones preliminares necesarias para trabajar con comodidad, como por ejemplo la configuración del teclado en castellano. Si no quiere perder tiempo buscando en un bosque de ventanas y tampoco tratar de hallar por tanteo en el teclado USA los caracteres que necesita, puede solucionarlo de manera rápida y sencilla a través de la misma consola de texto:

```
setxkbmap es
```

Tras esto podrá utilizar sin ningún problema el idioma de Cervantes en el teclado de su sistema Helix.

Como se ha mencionado, a diferencia de otras distribuciones compiladas para tareas de seguridad o reparación de sistemas, Helix está enfocada a cometidos de investigación forense. Para facilitar su uso los desarrolladores han decidido que sea lo suficientemente pequeña para caber en un CD-ROM. Aunque la mayor parte de los ordenadores que el investigador encontrará en cualquier escenario de intervención están basados en arquitecturas x86 y disponen de unidades de lectura DVD/CD-ROM, alguno de ellos podría ser lo bastante antiguo como para tener lector CD-ROM, pero no DVD-ROM. Con una imagen ISO y utilidades especiales, Helix3 Pro™ también puede grabarse en una llave USB y conseguir que arranque desde ella, a condición de que la BIOS del ordenador admita el inicio del sistema desde unidades externas.

### 9.4.2 Algunos aspectos de interés forense en Helix

Siempre que el investigador vaya a trabajar con medios removibles y estos lo admitan, debe asegurarse de que los mecanismos de protección (palancas de plástico, lengüetas, etc.) estén accionados para excluir un montaje accidental del

soporte o, peor aún, la posibilidad de confusión entre unidades de origen y destino. La investigación forense no es inmune a la Ley de Murphy. Por defecto Helix monta todas las unidades en modo estricto de solo lectura. Pero si tiene bloqueadores de escritura y no quiere perder tiempo explicando pormenores técnicos delante del tribunal, utilícelos.

A la hora de habilitar una unidad de destino para su imagen forense o los elementos de evidencia recolectados, montándola para ello con privilegios de escritura, el investigador deberá asegurarse de que está usando el comando “mount” con el medio de almacenamiento y sobre todo no equivocarse a la hora de seleccionar el dispositivo /dev/sdX. Si se confunde y monta el dispositivo sospechoso puede arruinar la evidencia aunque no toque los contenidos del disco duro. El solo acto de montar una partición con *journaling* en modo escritura ya cambia su *hash* al introducir variaciones mínimas en los archivos que registran el estado de las transacciones atómicas.

#### 9.4.3 Helix en una máquina virtual

Si el investigador tiene instalado en su estación de trabajo algún software de virtualización como VMware o Virtual Box, puede probar los CD autoarrancables de Helix sin tener que reiniciar el ordenador. Lo único que para ello necesita disponer de las imágenes ISO. Basta crear mediante el asistente de VMware una máquina virtual para Ubuntu, asignar una determinada cantidad de memoria RAM y seleccionar la imagen ISO de Helix en lugar de un disco duro virtual (figura 9.16).

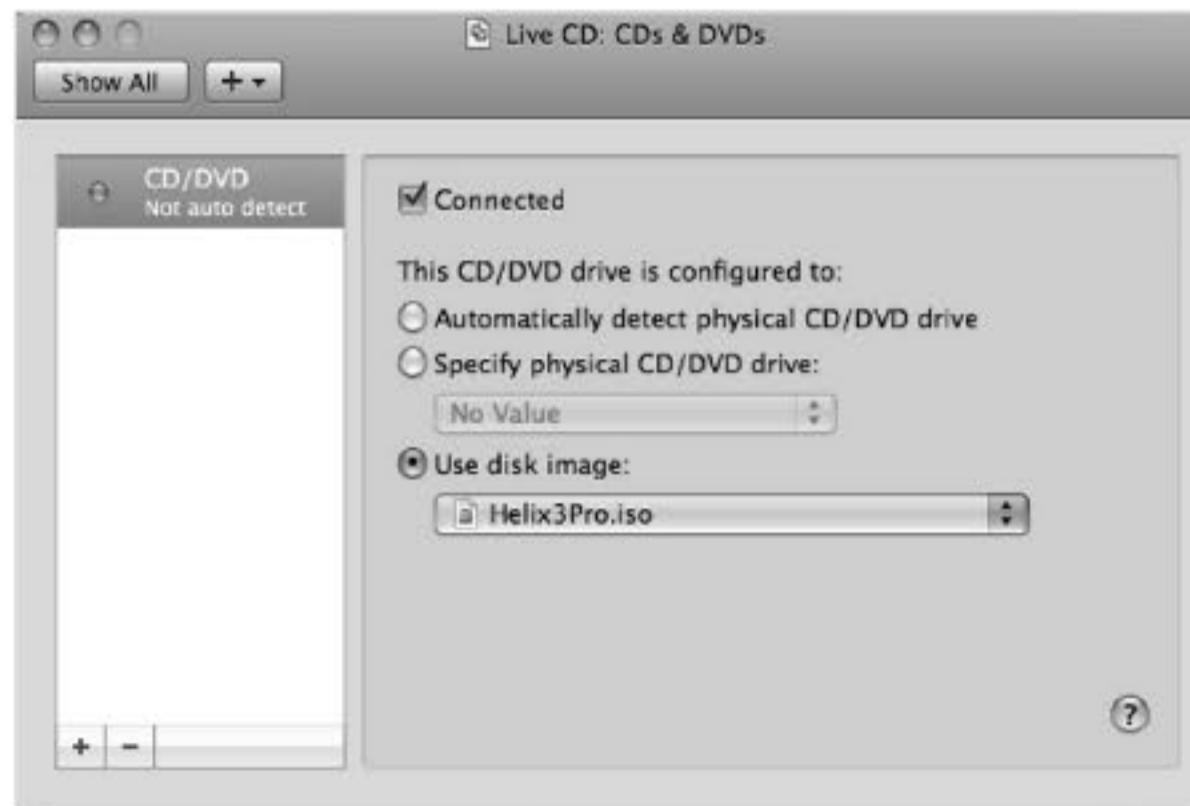


Figura 9.16. Helix sobre VMware

El investigador debe tener en cuenta, sin embargo, que las cosas no son del todo iguales a través de un sistema virtualizado. Para hacerse una idea de las prestaciones efectivas de Helix en su modalidad de CD autoarrancable, tarde o temprano tendrá que probarlo directamente sobre el hardware.

## Capítulo 10

# HERRAMIENTAS SOFTWARE

A lo largo de la presente obra el lector ha tenido ocasión de conocer las principales herramientas de software utilizadas en el campo de la Informática Forense. El método ha consistido en ir presentándolas a medida que el apartado concreto lo requería. Así, al hablar de la adquisición de imágenes forenses se habló de dd, AIR, Linen. Cuando se trató del análisis de evidencias electrónicas o la recuperación de archivos borrados el lector tuvo la oportunidad de conocer TSK o GetDataBack, y así sucesivamente. También se ha hablado de EnCase y FTK, las herramientas comerciales de más amplio uso en el ámbito de la investigación digital. El presente capítulo obedece a la necesidad de completar el cuadro con algunas aplicaciones adicionales que pueden ser de utilidad para el investigador. Entre ellas figuran varias distribuciones Linux y los programas de virtualización VMware y VirtualBox. Todos estos productos, algunos de código libre, otros comerciales, disponen de funcionalidades interesantes relacionadas con la economía, el ahorro de tiempo o de carácter didáctico que se irán comentando junto con una breve descripción del software.

Las distribuciones Linux que se mencionan en el presente artículo están disponibles para descarga en la página web <http://www.distrowatch.com>.

### 10.1 DISTRIBUCIONES LINUX

#### 10.1.1 Backtrack

Backtrack, basada en Ubuntu y actualmente en su versión 5, no es propiamente una distribución para investigadores forenses, sino una herramienta de

seguridad cuyo cometido consiste en verificar la resistencia frente a intrusiones de un sistema informático mediante técnicas de *hacking* ético. Sin embargo también contiene la mayor parte de las herramientas necesarias en un caso de investigación (dd, ddrescue, scalpel, TSK+Autopsy, netcat, etc.), pudiendo servir perfectamente como entorno de base para una adquisición forense o la previsualización de datos en el ordenador de un usuario sospechoso. Backtrack es una herramienta de código libre y puede obtenerse en forma de imagen ISO en la página web del desarrollador (<http://www.backtrack-linux.org>). A partir de la versión 5 está disponible para arquitecturas de 32 y 64 bits. Además de utilizarla como Live-CD puede ser instalada en un ordenador portátil, con el objeto de lograr un mayor rendimiento del software y ahorrar tiempo en el arranque del sistema operativo.



Figura 10.1. Interfaz gráfica de Backtrack con varios menús desplegados

La verdadera utilidad de Backtrack no reside en su arsenal forense, sino en sus capacidades para la monitorización y el análisis de comunicaciones, que la convierten en una herramienta insustituible a la hora de investigar casos de intrusión o realizar operaciones de vigilancia encubierta en entornos de red. Así por ejemplo Backtrack lleva consigo la suite Aircrack-ng para verificación y comprobación de seguridad en redes inalámbricas, integrada con las aplicaciones de gestión WiFi del sistema operativo y acompañada de controladores para gran número de tarjetas inalámbricas disponibles en el mercado, cuyo hardware es reconocido por Backtrack nada más insertarlas en el conector USB.

En Windows –y también en cualquier distribución popular de Linux como Ubuntu, CentOS o Mint– poner un interfaz Ethernet en modo promiscuo o comutar al modo monitor de las tarjetas WiFi, algo imprescindible para captar todos los paquetes inalámbricos que los puntos de acceso intercambian con los adaptadores del entorno, dista de ser una tarea trivial. A menudo para conseguir esta funcionalidad es necesario instalar controladores especiales y seguir complicados procedimientos de ajuste.

En Windows el modo monitor no está soportado por el *driver* de la mayor parte de los dispositivos, incluyendo los que vienen incorporados en la placa principal del ordenador, y ello por obvias razones. El fabricante quiere que el usuario disfrute navegando por Internet y se ocupe de sus propios asuntos. No le interesa que ande por ahí interceptando el tráfico inalámbrico de sus vecinos y enterándose de lo que estos hacen en la red. Backtrack por el contrario viene ya preparada para utilizar todos sus interfaces de conexión a pleno rendimiento. No es necesaria ninguna manipulación adicional para pasar una tarjeta de red al modo promiscuo o utilizar un interfaz inalámbrico en modo monitor. Bastan unos sencillos comandos o incluso iniciar Wireshark, y el propio software de monitorización comuta automáticamente al modo monitor en cuanto el usuario selecciona el interfaz.

Backtrack inicia siempre en modo superusuario. Todos los recursos del sistema se hallan disponibles sin que uno tenga que estar otorgándose a cada momento permisos especiales mediante el comando “*sudo*”, ya sea para habilitar interfaces de red, ponerlos en modo monitor, acceder a volúmenes y medios de almacenamiento, etc. En estas condiciones la capacidad para provocar daños en el sistema o en los soportes de datos investigados, por culpa de una equivocación o manejo inadecuado, es directamente proporcional al poder sin límites que se adquiere mediante el arranque del ordenador en modo *root*. Por la misma razón conviene que el investigador extreme las precauciones a la hora de montar una partición y siempre que pueda lo haga exclusivamente a través de bloqueadores de escritura.

### 10.1.2 Knoppix

Knoppix es una distribución Live-CD muy popular entre técnicos, estudiantes y aficionados a las tecnologías de la información y al software libre con un nivel de conocimientos alto. Fue desarrollada en 2003 por Klaus Knopper, quien todavía se encarga de mantenerla y coordinar las nuevas versiones. Knoppix está disponible tanto en CD como en DVD. Puede utilizarse como Live-CD y también instalarse en el disco duro. Incluye, además de un número de herramientas útiles para tareas de reparación y rescate de datos, los programas habituales para realizar

cualquier cometido en un entorno de informática de gestión: entorno de ventanas KDE o LXDE (en las versiones recientes), OpenOffice, lenguajes de programación y la herramienta de retoque fotográfico de código libre GIMP.

El verdadero interés de Knoppix no reside pese a ello en su utilidad para fines forenses, sino más bien en un valor didáctico que tiene que ver con su intención inicial de constituir la primera distribución popular de Linux pensada para ejecutarse desde un CD. Esto permite al usuario probar el sistema operativo y dar sus primeros pasos en Linux, con una funcionalidad completa en cuanto a reconocimiento del hardware y ejecución de los programas habituales, sin tener que habilitar particiones ni instalar nada, y sin tener que modificar la configuración de su portátil o sistema de sobremesa (típicamente Windows).

Precisamente esta funcionalidad de detección de hardware en el arranque del sistema, que puede ser una ventaja para el estudiante y el técnico, hace arriesgado emplear Knoppix en un entorno de investigación forense, a no ser que se utilicen bloqueadores de escritura. No existen garantías de que Knoppix acceda a una partición con *journaling* de un modo conveniente para el investigador, es decir, sin realizar pequeñas modificaciones que alterarían el *hash* del soporte de datos. Aunque dichos cambios son mínimos y no afecten al contenido de los datos guardados en un disco duro, vaya después a explicárselo usted al abogado de la parte contraria.



Figura 10.2. Knoppix (Fuente: Distrowatch)

El lector se quedaría con una impresión inadecuada si de lo dicho con anterioridad dedujera que Knoppix es una especie de escaparate cuyo cometido consiste en animar a los usuarios con ganas de saber a que den el salto definitivo de Windows al mundo del software libre. En realidad Knoppix es un entorno de aprendizaje completo para iniciarse y adquirir maestría en el dominio de Linux como usuario avanzado. Esto incluye no solamente el manejo de aplicaciones de productividad estándar, como procesadores de texto o programas de retoque fotográfico, sino también sofisticadas y potentes herramientas de manipulación de datos y reparación del sistema.

Entre la abundante documentación sobre Knoppix merece la pena destacar el libro de Kyle Rankin *Knoppix Hacks* (O'Reilly, 2007), compendio de técnicas orientadas no solo al aprendizaje sino a la resolución de problemas de todo tipo, mediante el cual el usuario

podrá adquirir una competencia auténticamente profesional en el manejo de Linux y utilizar las herramientas incluidas en el Live-CD Knoppix para llevar a cabo operaciones de alto nivel, que en determinadas circunstancias pueden llegar a tener un interés forense: reparar estructuras de datos en sistemas Linux y Windows, habilitar un cliente de red, reemplazar servidores web o cortafuegos en caso de emergencia, llevar a cabo auditorías de seguridad en entornos de red o escanear un sistema Windows en busca de virus y troyanos.

Knoppix está disponible en la página web de su desarrollador Klaus Knopper: <http://www.knoppix.org>.

### 10.1.3 SystemRescueCD

SystemRescueCD es un sistema Linux para tareas de rescate disponible como Live-CD o llave USB autoarrancable y diseñado con la idea de reparar sistemas después de un *crash*. Permite realizar tareas típicas de administrador como editar o reparar particiones de disco duro sin tener que recurrir a los discos de instalación originales. Para ello dispone de todas las herramientas Linux de sistema (gestor de particiones, partimage, fstools) así como de utilidades básicas (editores de tareas, herramientas de red, etc.), todo ello condensado en una imagen ISO cuyo tamaño es la mitad de una distribución Live-CD normal, y que cabe perfectamente en un CD-ROM.

En SystemRescueCD el investigador no encontrará más que lo básico: nada de aplicaciones de ofimática, sino editores de texto y un simple visor de documentos; y tampoco sofisticados entornos gráficos, sino tan solo el espartano gestor de ventanas XFCE. Con este enfoque minimalista, orientado a reducir en lo

posible el impacto en la solicitud de recursos, SystemRescueCD, además de reconocer la mayor parte de los sistemas de archivos existentes en la actualidad, incluye un número de herramientas que permiten utilizarla para fines de investigación forense: dd, ddrescue, TSK, Testdisk y el *driver* Ntfs3g para lograr acceso de lectura y escritura a particiones NTFS Windows.

Las funcionalidades de reconocimiento de hardware de SystemRescueCD son muy limitadas, quedando reducidas en la práctica a la capacidad del *kernel* y sus módulos para detectar dispositivos básicos en el momento del arranque. Pero esto, además de responder a la intención del desarrollador que no es otra que la de crear una herramienta para tareas de rescate, lejos de ser un inconveniente constituye también una ventaja para los fines que persigue un investigador forense que utiliza SystemRescueCD, por ejemplo, como herramienta de circunstancias, cuando no tiene a mano su equipo habitual. No solamente ahorra tiempo en el arranque del sistema sino que además, al no disponer de montaje automático de volúmenes, no implica riesgo de alteración accidental de una partición con *journaling*.

SystemRescueCD se puede descargar desde <http://www.sysresccd.org>.

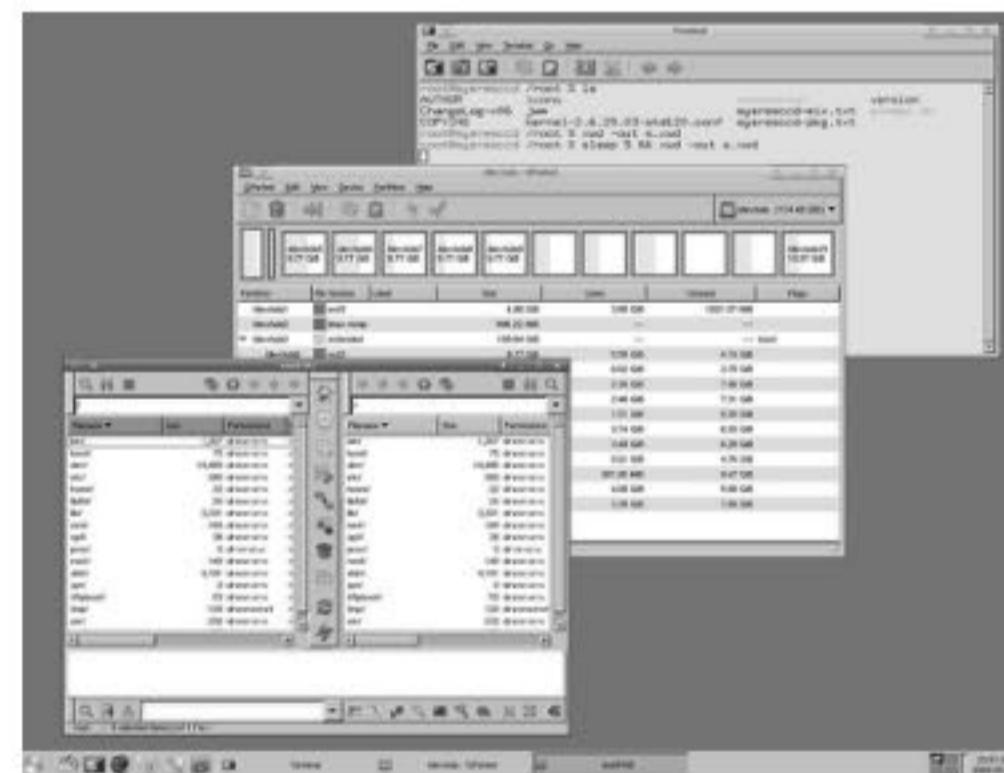


Figura 10.3. SystemRescueCD y XFCE (Fuente: Distrowatch)

### 10.1.4 CAINE

CAINE (*Computer Aided Investigative Environment*) es una distribución Live-CD Linux italiana gestionada por el experto en Informática Forense Nanni Bassetti, y está provista de un entorno gráfico desde el cual el investigador puede

acceder con facilidad a todas las herramientas que le permiten llevar a cabo el ciclo de trabajo completo: previsualización, adquisición, análisis y realización de informes. Se trata de un proyecto 100% *Open Source*, con documentación y fuentes disponibles de manera que los trabajos de desarrollo y mantenimiento puedan ser llevados a cabo por cualquier persona interesada. De manera análoga a Helix, CAINE dispone de un interfaz llamado WinTaylor (figura 10.5) que corre sobre Windows y está desarrollado en VisualBasic 6, con el objeto de asegurar la compatibilidad con sistemas Microsoft.

WinTaylor incluye una batería de herramientas utilizadas habitualmente para adquisiciones forenses en vivo: FTK Imager, Testdisk, PSTools, utilidades para hacer volcados de memoria RAM, etc. Fiel a la filosofía que orienta el desarrollo de CAINE, el código de WinTaylor también es libre y está puesto a disposición de la comunidad por si alguien desea introducir mejoras o simplemente para evitar que el proyecto se pierda en caso de que sus actuales administradores no puedan seguir gestionándolo.

WinTaylor, además de una estructura tabular y de pestañas para las diferentes funciones del software forense parecidas a la de Helix, dispone de un generador de informes que extrae en un archivo de texto la relación de programas utilizados junto con sus marcas de tiempo correspondientes. También permite trabajar en línea de comando e incluye una versión actualizada de las herramientas Sysinternals, utilidades para el cálculo de *hashes* y una aplicación para realizar instantáneas con fines documentales.

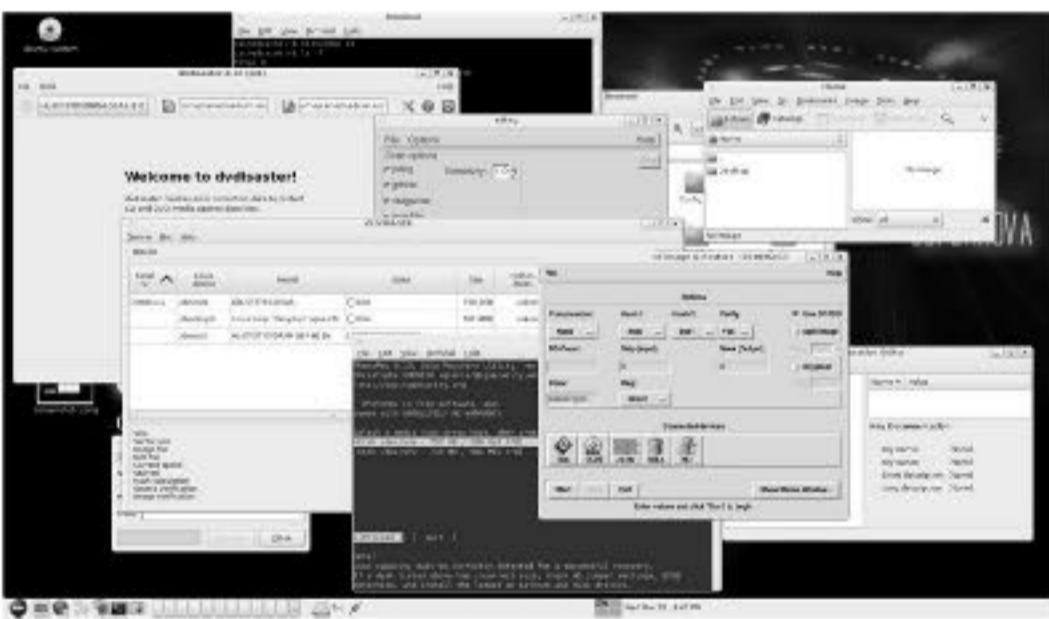


Figura 10.4. CAINE

En cuanto al cumplimiento de requisitos típicos de las investigaciones forenses, los desarrolladores prestan especial atención a este apartado mediante la inclusión de “rbfstab”, una utilidad que se activa automáticamente durante el arranque del sistema en modo Linux o al conectar un soporte de datos. Funciona

escribiendo al archivo /etc/fstab las entradas necesarias para un montaje de todas las particiones en modo de solo lectura, con lo cual se asegura un acceso a los datos en condiciones de seguridad y de auténtica higiene forense. CAINE dispone de numerosos *scripts*, activados con el ratón desde el navegador de archivos Nautilus, que simplifican la tarea de análisis de los archivos encontrados. Con estos *scripts* resulta posible extraer una cantidad considerable de información en archivos de texto: historial de Internet, registro de Windows, metadatos Exif, archivos borrados, etc.

Desde el mismo interfaz gráfico, a través de la utilidad “mounter”, se puede así mismo controlar el montaje de particiones en modo lectura y montar dispositivos conectados en modo escritura para trasladar a él elementos de evidencia y resultados del análisis. CAINE también dispone de un *script* llamado “Identify iPod Owner” que permite examinar de manera rápida y simple los datos de un iPod conectado al sistema (nombre de usuario, número de serie, etc.), así como información relacionada con la configuración del software iTunes utilizado para sincronizar el dispositivo.

CAINE y WinTaylor están disponibles para descargar en <http://www.caine-live.net/>.



Figura 10.5. WinTaylor

### 10.1.5 Slackware

Slackware (<http://www.slackware.com>) fue una de las primeras distribuciones Linux, muy anterior a las compilaciones populares y amigables con el usuario de nuestros días, puesto que fue creada por Patrick Volkerding en 1993. Son varias las razones que explican esta persistencia: simplicidad de diseño,

fidelidad a los conceptos tradicionales de Unix y una estabilidad a prueba de bomba que convierten a Slackware en una opción favorita para numerosos técnicos de sistemas a la hora de montar servidores y estaciones de trabajo para aplicaciones de ingeniería. Slackware, que en el momento de escribir estas líneas se encuentra en su versión 13.37, está destinado a aquellos usuarios a los que les gusta aprender y disfrutan configurando su sistema para hacer estrictamente lo que les interesa sin ninguna concesión a funciones adicionales sobre las que el administrador carezca de control. Por todos estos motivos es de esperar que esta distribución, pese a su diseño espartano a base de paquetes comprimidos LZH, y su carencia de un interfaz gráfico de instalación (figura 10.6), continúe siendo utilizada aún durante los próximos años.

Si se ha tomado la decisión de incluir en este libro un apartado sobre Slackware ello se debe a que sus posibilidades de configuración granular permiten construir una estación de trabajo ideal para finalidades de investigación forense, eliminando o modificando de manera controlada y consciente todos aquellos componentes de software que hacen posible el montaje automático de particiones y volúmenes. De igual manera, al organizar un entorno de trabajo forense interesa que los recursos del ordenador estén disponibles para las herramientas: que no haya procesos innecesarios sustrayendo ciclos de microprocesador a aquellas tareas cuya ejecución se considera prioritaria: búsqueda de caracteres, *cracking* de contraseñas, etc. También desearíamos que en el disco duro quede la mayor cantidad de espacio posible para guardar datos de trabajo relacionados con una investigación forense (imágenes en *bitstream*, archivos rescatados, datos de análisis, *rainbow tables* para ataques de fuerza bruta a contraseñas, etc.). La instalación de un sistema completo con todo lo necesario para trabajar en un cometido de investigación forense, pero nada más aparte de eso, no tiene por qué ocupar más de 6 GB.

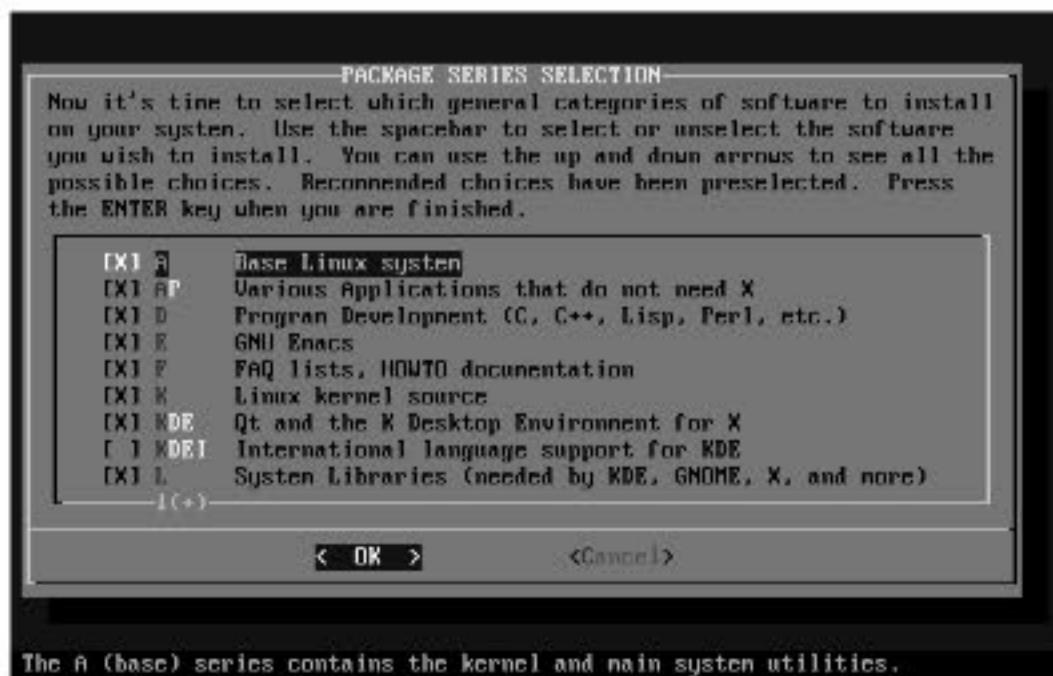


Figura 10.6. Selección de paquetes en Slackware 13.0

La instalación de Slackware no solo permite elegir entre grupos de paquetes, sino entre paquetes concretos dentro de un mismo grupo. En principio necesitaríamos, además del sistema base, las librerías dinámicas, el sistema de ventanas X Windows y posiblemente también el entorno de desarrollo y programación –compiladores, *make*, etc.– y las fuentes actualizadas del *kernel*, por si fuera necesario recompilarlo para incorporar alguna característica funcional nueva –soporte para periféricos, o acceso a determinados sistemas de archivos–. Podemos prescindir, por otro lado, de voluminosos y complejos editores de texto como Emacs así como de TeX y la serie TCL. También sería conveniente dejar al margen a otros grandes consumidores de espacio en disco como GNOME y KDE. Deberíamos reemplazarlos por el más liviano y económico en recursos XFCE.

En el capítulo 5 se habló de la problemática relacionada con el montaje automático de particiones en Linux, mencionándose la posibilidad de inhabilitar este mecanismo en Slackware. Ahora vamos a ver de qué forma se consigue evitar la interacción entre udev, HAL y d-messagbus de manera que en el arranque Linux no haga nada que pudiera llegar a comprometer la integridad de los datos forenses. Para ello es necesario modificar los permisos de ejecución de los *daemons* de estos módulos para evitar su carga en el inicio del sistema. Gracias a estas modificaciones no necesitaremos bloqueadores de escritura para proteger la evidencia de cualquier posibilidad de acceso no autorizado. Y por si fuera poco los cambios son fácilmente reversibles.

Desde la versión 7.0 de Slackware, los *scripts* de arranque están estructurados de tal manera que existe un directorio por cada nivel de ejecución –recordemos que en Linux/Unix los niveles de ejecución determinan de modo esencial la funcionalidad global del sistema: desde el nivel 0 (apagado) hasta el 6 (reinicio del sistema) pasando por el 1 (modo simple monousuario), los 2, 3 y 5 (normal, red y con capacidad para ejecutar el entorno gráfico X, todos ellos multiusuario), y finalmente el nivel 4 (en reserva y sin asignar)–. Cada uno de estos directorios contiene los *scripts* necesarios para poner al sistema en el nivel de ejecución correspondiente. Un *script* –de modo análogo a los antiguos archivos CONFIG.SYS y AUTOEXEC.BAT para la configuración inicial de MSDOS– no es más que un guión o lista de comandos que realiza una secuencia automatizada de tareas.

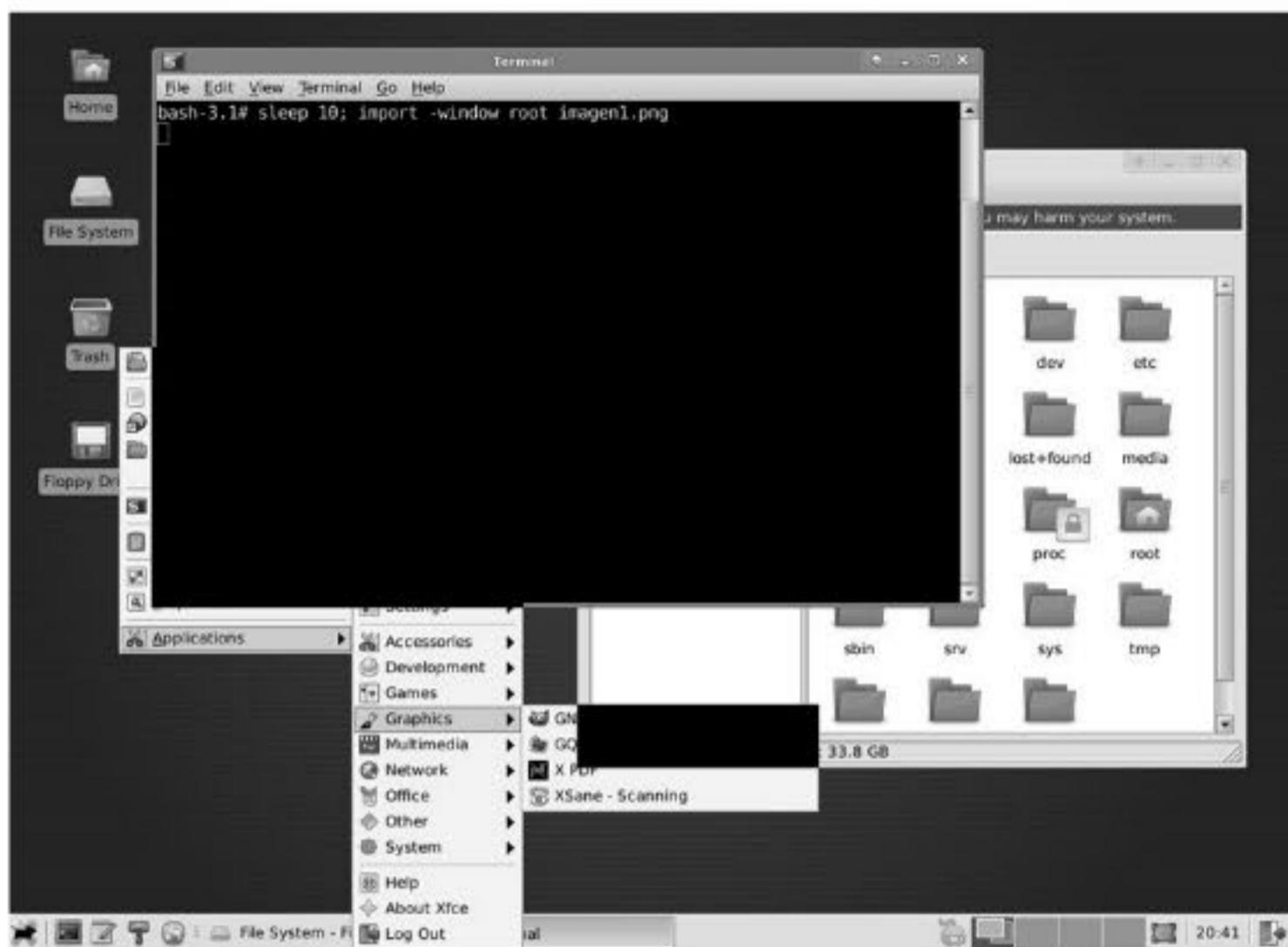


Figura 10.7. Entorno de ventanas XFCE sobre Slackware

Una vez establecido el nivel de ejecución o *runlevel* mediante inittab y ejecutados los *scripts* globales de arranque para tareas básicas de inicialización del sistema, actúa el *script* por defecto de Slackware, /etc/rc.d/rc.M, cuya misión consiste en dar vida a diversos servicios y *daemons* del sistema. Examinando en detalle el archivo que contiene el *script*, por ejemplo de este modo:

```
root@forensics: ~# cat /etc/rc.d/rc.M
```

... hallaremos varias líneas de un formato parecido a este:

```
.....  
if [ -x /etc/rc.d/rc.pcmcia ]; then  
./etc/rc.d/rc.pcmcia start  
.....
```

La sentencia condicional quiere decir que el *script* se ejecutará desde rc.M únicamente si posee permisos para ello. Lo que en este caso se quiere evitar es que el sistema acceda a soportes de datos sin que su propietario lo ordene de manera

explícita. Para ello lo único que hay que hacer es modificar los permisos de ejecución de los *scripts* que inician HAL y d-messagbus durante el arranque.

```
root@forensics: ~# chmod 644 /etc/rc.d/rc.hald  
root@forensics: ~# chmod 644 /etc/rc.d/rc.messagebus
```

Esto no impedirá que los dispositivos sean detectados durante el arranque. Si el investigador está utilizando una versión reciente de Slackware (como por ejemplo la 12 o la 13), su sistema estará construido sobre el Kernel 2.6 y los elementos del hardware figurarán en el listado del comando dmesg. La inutilización de HAL y d-messagbus lo único que hace es impedir el montaje automático de los soportes correspondientes. Los cambios, a no ser que los procesos de ejecución de los respectivos *daemons* sean detenidos de manera explícita, serán efectivos tras el reinicio del sistema. Con la misma facilidad se puede hacer que el cambio sea reversible y restablecer todos estos automatismos durante el arranque.

## 10.2 VIRTUALIZACIÓN

Hay varias razones por las que un investigador forense podría llegar a interesarse por la virtualización. Tal vez quiera probar distribuciones de Linux sin tener que reiniciar el ordenador, o un software nuevo incompatible con la configuración actual de su estación de trabajo. Quizás necesite disponer de un entorno aislado de su sistema (una *sandbox*) para el análisis de *malware*, utilizar MS-Word o Excel sobre un Windows virtual para realizar su informe mientras trabaja con TSK en su *host* Linux o por el contrario servirse de aplicaciones Linux en una máquina virtual mientras utiliza el *host* como plataforma principal para llevar a cabo análisis con EnCase o FTK. O posiblemente desee familiarizarse con los conceptos fundamentales y las tecnologías en las que se apoya eso que llaman “la nube”, implementada principalmente a base de máquinas virtuales, y que con su progresivo dominio de Internet y el mundo de las redes plantea nuevos problemas a la Informática Forense separando la propiedad de los datos de su gestión e impidiendo el acceso directo al hardware por parte de investigadores y administraciones de sistemas.

Las tecnologías de virtualización distan de ser novedosas, puesto que hace décadas los grandes ordenadores del tipo *mainframe* ya las utilizaban para aprovechar mejor las capacidades de multitarea de los sistemas operativos y los recursos del hardware. Para hacer uso de ellas en su estación de trabajo forense el investigador dispone de un número de productos populares para elegir, tanto comerciales como de código libre.



Figura 10.8. Windows XP sobre Windows Vista mediante VMware Workstation

### 10.2.1 VMware

VMware (<http://www.vmware.com>) es un software desarrollado por VMware Inc., filial de EMC Corporation, que con una gama de productos concentrada en este negocio principal suministra al mercado la mayor parte de las soluciones de virtualización para plataformas con arquitectura x86 de 32 y 64 bits. Además de artículos de pago como VMware Workstation –que viene a ser prácticamente el estándar en este sector de la industria informática–, dicha gama de producto incluye aplicaciones gratuitas como VMware Server y VMware Player. El software de VMware puede funcionar tanto en Linux como en Windows y en plataformas Apple OS X provistas de CPU Intel, para las que existe una versión específica denominada VMware Fusion.

A diferencia de los emuladores del tipo Wine, que traducían las API Windows a llamadas de sistema para que con diversas limitaciones pudieran ejecutarse sobre un *kernel* Linux, VMware es un sistema de virtualización completo por software que simula un entorno físico con unas características hardware determinadas (CPU, memoria, disco duro, BIOS, interfaces de red, tarjeta de sonido y hasta conexiones USB). En resumidas cuentas, se trata de engañar al sistema operativo huésped para que crea que está accediendo al hardware cuando en realidad lo único que hace es manejar una máquina virtual. El verdadero control de los recursos físicos lo tiene en todo momento el sistema anfitrión, y es VMware

quién pasa a este todas las solicitudes de empleo de un determinado recurso –por ejemplo un periférico o el interfaz de red– para que lo gestione a bajo nivel.

Pero salvo esta dependencia común de los recursos básicos, administrada por un núcleo MS-Windows o un *kernel* Linux, para todo lo demás parecerá que el usuario está manejando máquinas diferentes, con su microprocesador, su memoria e incluso sus propios interfaces de red. El vínculo entre los dos sistemas depende de la existencia de carpetas compartidas para intercambio de datos y de las decisiones tomadas por el usuario a la hora de configurar la conexión en red entre el sistema anfitrión y el *host*. Sobre un entorno VMware pueden funcionar varias máquinas virtuales de manera simultánea, siendo posible montar una red virtual completa, incluso con sistemas operativos diferentes, y de este modo probar herramientas de seguridad, monitorización y adquisición forense remota sin tener que habilitar para ello otras máquinas.

Conviene mencionar que a diferencia de otras soluciones similares como Virtual PC de Microsoft, VMware no es un emulador sino un software de virtualización en el sentido estricto. Su eficacia procede en gran parte de su capacidad para aprovechar el modo virtual protegido de una CPU x86, que divide el microprocesador en varias máquinas diferentes pudiendo cada una de ellas ser manejada por un sistema distinto. En lugar de traducir las instrucciones a llamadas de sistema que posteriormente se ejecutan en el hardware, lo que VMware hace es ejecutar la mayor parte de sus instrucciones directamente sobre el sistema físico. El rendimiento del sistema virtual es variable en función de las características del hardware sobre el que se ejecuta y de los recursos virtuales asignados a través de VMware (CPU, RAM, disco duro) en el momento de crear la máquina virtual.

### 10.2.2 VirtualBox

El software de virtualización VirtualBox fue creado originalmente por la empresa alemana Innotek GmbH y posteriormente adquirido por Oracle quien lo incorporó como un elemento más de su gama de productos de virtualización para arquitecturas Intel y AMD 64. Como en el caso de VMware, VirtualBox funciona estableciendo un entorno virtual que permite el funcionamiento de uno o varios sistemas invitados sobre otro anfitrión. Entre los sistemas soportados por VirtualBox como anfitrión se encuentran los habituales en el mundo de la informática de usuario (MS-Windows, GNU/Linux y OSX) y otros de tipo más profesional (Solaris), sobre los cuales existe la posibilidad de virtualizar gran número de plataformas, incluyendo versiones especiales de Unix, sistemas obsoletos como OS2/Warp e incluso fósiles informáticos como MSDOS.

VirtualBox es gratuito y se halla incluido en los repositorios de las distribuciones Linux habituales. Añadirlo a Ubuntu es tan simple como teclear “sudo apt-get aptitude virtualbox” en una consola de texto o seleccionarlo en el interfaz gráfico de gestión de paquetes. Para instalarlo en Windows el lector tendrá que ir a la página de Oracle donde está disponible para descarga (<http://www.virtualbox.org>). Una dificultad con la que el investigador puede encontrarse, sobre todo si su principal interés reside en el análisis de *pendrives* o discos duros externos, es que la versión gratuita de VirtualBox carece de soporte para el interfaz de conexión USB. Para superar este obstáculo, a no ser que quiera dar un rodeo a través de carpetas compartidas entre la máquina virtual y el sistema anfitrión, deberá descargarse desde la página de Oracle una versión especial de VirtualBox con extensiones para USB y otras características funcionales que no es gratuita, salvo para usos particulares y de evaluación del producto.



Figura 10.9. Windows Vista sobre VirtualBox

### 10.2.3 Listado de herramientas

Para terminar el presente capítulo se expone un listado de las principales herramientas mencionadas a lo largo del libro, junto con un breve resumen de sus características y el tipo de software. Las referencias a empresas y páginas web no se incluyen por diversas razones. Algunas de estas aplicaciones están descontinuadas o pertenecían a casas que han sido adquiridas por otras compañías durante los últimos años. Para obtener información actualizada acerca de estas

herramientas así como sobre su estado actual de desarrollo, modalidades de licencia, sitios de descarga y otras particularidades se recomienda llevar a cabo las correspondientes búsquedas en Google.

Herramienta	Función	Plataforma	Código
AccessData FTK / Imager	Suite integrada de análisis forense	Windows	Propietario
Adepto	Adquisición forense	Linux	Libre
Adobe Photoshop	Software de retoque fotográfico	Windows/OSX	Propietario
AIR	Adquisición	Linux	Libre
Android SDK	Kit de desarrollo de Android	Linux/Windows/OSX	Libre
Autopsy + The Sleuth Kit	Análisis forense	Linux/Windows	Libre
Backtrack	Distribución para auditoría de redes	Linux	Libre
CAINE	Distribución de Informática Forense	Linux/Windows	Libre
Captain Nemo	Montaje de imágenes forenses	Windows	Propietario
Chkrootkit	Detector de rootkits para sistemas Linux	Linux	Libre
Cygnus	Entorno Unix para Windows	Windows	Libre
dc3dd	Adquisición forense	Linux/Windows	Libre
Dcfldd	Adquisición forense	Linux/Windows	Libre
dd	Adquisición forense	Linux/Windows	Libre
ddrescue	Adquisición forense	Linux/Windows	Libre
Disk Investigator	Búsqueda de cadenas de caracteres	Windows	Libre
EasyRecovery Professional	Recuperación de archivos borrados	Windows	Propietario
EnCase	Suite integrada de análisis forense	Windows	Propietario
Evidor	Búsqueda de cadenas de caracteres	Windows	Propietario
Exiftool	Ánalisis de metadatos de archivos gráficos	Linux/Windows	Libre
FileDisk	Montaje de imágenes forenses	Windows	Propietario
FOCA	Ánalisis de metadatos	Windows	Propietario

Herramienta	Función	Plataforma	Código
Foremost	<i>Data carving</i>	Linux/Windows	Libre
Galleta	Análisis de <i>cookies</i>	Linux/Windows	Libre
GIMP	Software de retoque fotográfico	Linux/Windows	Libre
Helix	Distribución de Informática Forense	Linux/Windows	Libre /Comercial
iehist	Análisis del historial de Internet	Windows	Libre
iPhone Backup Extractor	Análisis de <i>backups</i> de iTunes	Windows	Propietario
iTunes	Herramienta de sincronización para dispositivos Apple	OSX/Windows	Propietario
Knoppix	Distribución Live-CD	Linux	Libre
Linen	Adquisición	Linux/DOS	Propietario
Md5deep	Cálculo recurrente de <i>hashes</i> MD5	Linux/Windows	Libre
Metadata Assistant	Análisis de metadatos	Windows	Propietario
Mount Image Pro	Montaje de imágenes forenses	Windows	Propietario
Paraben's E-Mail Examiner	Análisis de archivos de correo electrónico	Windows	Propietario
Pasco	Análisis del archivo INDEX.DAT	Linux/Windows	Libre
Ping	Trazabilidad de redes	Linux/Windows	Libre
PsTools	Análisis de procesos MS-Windows	Windows	Propietario/Gratis
Rifiuti	Análisis de la papelera de Windows	Linux/Windows	Libre
Rkhunter	Detector de <i>rootkits</i> para sistemas Linux	Linux	Libre
R-Studio	Recuperación de archivos borrados	Windows	Propietario
Runtime DiskExplorer	Análisis de particiones FAT y NTFS	Windows	Propietario
Runtime GetDataBack	Recuperación de archivos borrados	Windows	Propietario
Scalpel	<i>Data carving</i>	Linux/Windows	Libre
SectorSpy	Búsqueda de cadenas de caracteres	Windows	Propietario

Herramienta	Función	Plataforma	Código
Slackware	Distribución Linux de tipo general	Linux	Libre
SMART	Suite integrada de análisis forense	Linux	Propietario
SQLitemanager	Cliente de bases de datos SQLite	Linux/Windows	Libre
Sysinternals	Análisis de procesos MS-Windows	Windows	Propietario/Gratis
SystemRescueCD	Reparación de sistemas	Linux	Libre
Testdisk + Photorec	Reparación & <i>Data carving</i>	Linux/Windows	Libre
Traceroute/Tracert	Trazabilidad de redes	Linux/Windows	Libre
VirtualBox	Virtualización	Linux/Windows	Libre/Propietario
VMware	Virtualización	Linux/Windows	Propietario
Whois	Búsqueda de información de dominios en Internet	Linux/Windows	Libre
Windows Registry Recovery	Análisis de archivos del Registro de Windows	Windows	Libre
Wireshark	Analizador de tráfico	Linux/Windows	Libre
X-Ways Forensics	Editor hexadecimal con funcionalidad extendida	Windows	Propietario
X-Ways Trace	Análisis del historial de Internet Explorer	Windows	Propietario

## CONCLUSIONES

### Capítulo 11

Llegados al final del trayecto doy por hecho que el lector tendrá ganas de preguntar tanto como lo que por razones de espacio hemos tenido que omitir. Por si fuera poco en estos momentos el campo de la Informática Forense está experimentando cambios que en pocos años podrían dejar obsoleto gran parte de lo aprendido. En este capítulo se intentará dar respuesta a algunas de las preguntas más esperables. Y la primera sería esta: ¿Ya está? ¿Esto es todo? ¿Realmente la labor del investigador consiste en llevar ante el estrado del juez unos elementos de evidencia digital con un grado de profesionalidad suficiente para que quede demostrado el mantenimiento de una cadena de custodia y el abogado de la parte contraria no se atreva a impugnar nuestro bien escrito y mejor fundamentado informe? ¿Para eso era necesario aprender todas esas nociones avanzadas sobre sistemas de archivos, tecnologías de almacenamiento de datos y funcionamiento de redes? ¿No habría bastado con un catálogo de buenas prácticas y la Ley de Enjuiciamiento Civil?

Esta no es una salida de simple trámite, sino la parte más difícil del libro. En ella el autor aspira no solamente a justificar lo que el lector paga por llevarse este libro a casa, sino a transmitir una impresión adecuada del significado que la Informática Forense y la investigación de medios digitales tienen en un mundo cada vez más permeado por las Tecnologías de la Información. En pocos años la civilización moderna ha terminado por ser más dependiente del software y las máquinas que lo procesan que del petróleo y los recursos naturales. Además de pasar revista al estado de esta disciplina, se hará referencia a algunas aplicaciones útiles fuera del ámbito legal, a los principales retos para el futuro y otros aspectos merecedores de interés.

### 11.1 ESCENARIOS Y APLICACIONES

#### 11.1.1 En el Juzgado

Existen numerosas situaciones en las que una investigación forense puede ser necesaria o al menos conveniente. Las más obvias tendrían lugar, como es lógico, ante un tribunal en el que se esté juzgando un caso civil o penal, donde la misión del informático forense consiste en prestar sus servicios profesionales en relación con una serie de cometidos importantes para el proceso:

- Aseguramiento de la cadena de custodia.
- Validación de resultados mediante el empleo de herramientas alternativas o consulta a especialistas en determinadas materias.
- Elaboración de notas e informes periciales.
- Exposición de opiniones técnicas y facultativas delante del tribunal.

#### 11.1.2 Investigaciones en organizaciones y empresas

Otro escenario de intervención serían las investigaciones internas dentro de las empresas como respuesta a incidentes de seguridad, casos de deslealtad de empleados, fraude o espionaje industrial. Antes de poner un asunto en manos de los tribunales, a la dirección le interesa establecer con la mayor exactitud posible tanto la causa como el desarrollo de los incidentes y por supuesto la identidad de las personas involucradas en los mismos. A partir de esta información se determinará la conveniencia de tomar las medidas que haga falta: una acción disciplinaria, llevar el caso ante los tribunales o desarrollar una estrategia de relaciones públicas con el objeto de mitigar posibles daños de imagen provocados por el suceso, etc.

Una investigación corporativa puede afectar a los más variados ámbitos, dependiendo de la actividad de la empresa, la naturaleza concreta de las acciones delictivas cometidas, la categoría funcional de las personas que intervienen en las mismas o son sus víctimas y muchos otros factores. He aquí varios ejemplos:

- Auditorías de seguridad.
- Ataques contra redes de ordenadores.
- Intrusión de personas no autorizadas en un sistema informático.

- Espionaje corporativo y robo de propiedad intelectual.
- Acoso moral en el trabajo.
- Uso indebido de los recursos de la empresa.
- Fraudes y estafas.
- Sustracción de información relacionada con los clientes y/o las actividades comerciales de la empresa.
- Filtración de documentos confidenciales.
- Vulneración de la Ley de Protección de Datos.
- Existencia de *botnets* que utilizan los ordenadores de la empresa como zombis para atacar redes o como repositorios de *warez* y pornografía infantil.

### 11.1.3 Particulares y compañías de seguros

El informático forense también puede prestar un valioso servicio en situaciones que tienen que ver con intereses legales particulares y familiares, como demandas de divorcio o litigios por custodia, testamentarias, adjudicación de propiedades y bienes inmuebles, etc. Del mismo modo el investigador puede ser útil a las aseguradoras en investigaciones relativas a fraudes o bajas fingidas, donde las pruebas de autenticidad de fotografías digitales y la búsqueda de información en Internet y redes sociales están a la orden del día.

### 11.1.4 Sector público y seguridad nacional

Los gobiernos no solamente son los mayores ofertantes de empleo y al mismo tiempo los custodios más importantes de datos personales pertenecientes a los ciudadanos. Entre las responsabilidades de la administración pública también figura la gestión de extensas redes de ordenadores y bases de datos de gran tamaño con información confidencial sobre asuntos diplomáticos, económicos y militares. La existencia de gran cantidad de nodos y personas autorizadas, combinada con la heterogeneidad de los sistemas y el software instalado, que incluye parques de *mainframes* y máquinas antiguas supervivientes de los primeros tiempos de la informatización, aumenta la probabilidad de que se produzcan incidentes, siendo necesario disponer de personal capacitado para ayudar a hacerles frente con eficacia.

Los gobiernos suelen constituir objetivo potencial de ataque por parte de otros países y agencias de espionaje extranjeras, bandas terroristas o grupos antisistema. No hace falta poner ejemplos: todos los días lo podemos ver en la prensa y en Internet. La ofensiva de *hackers* rusos contra Chechenia en 2008 y el gusano Stuxnet deberían bastar para colmar cualquier necesidad de sensibilización del público a este respecto. Mediante una investigación profesional que determine causas y efectos el informático forense puede ayudar a poner en marcha escenarios de inteligencia y contraespionaje, contribuyendo así de manera decisiva a la seguridad nacional en un mundo que, arrastrado por el progreso técnico y la globalización, avanza sin resuello hacia el futuro.

## 11.2 OBSTÁCULOS

A lo largo de la obra se ha hecho mención a las diversas situaciones que pueden poner en peligro la evidencia proporcionando argumentos para una impugnación de la misma delante del tribunal –montaje automático de particiones con *journaling*, alteraciones de los datos, divagaciones caprichosas ante el juez–. Además de la negligencia existen otros enemigos, y no están precisamente en la sala de vistas sino al otro lado de la frontera legal. Muy probablemente hayan actuado ya mucho antes de que los resultados de la investigación entren en dependencias de la Administración de Justicia. Los intentos por parte de un usuario sospechoso de eliminar información comprometedora pueden dar que hacer al investigador forense. Es conveniente que este se halle al tanto de los trucos de los que un sospechoso –sobre todo si posee cierto nivel de conocimientos informáticos– puede llegar a servirse, preventivamente o *in extremis* al enterarse de la inminencia de un registro, para evitar verse incriminado por la prueba digital.

### 11.2.1 Destrucción intencionada de la evidencia

Un usuario que deseé destruir la evidencia generada por una utilización indebida de equipos informáticos, o al menos dificultar la tarea del investigador durante la adquisición y el análisis forense, puede llevar a cabo diversas acciones destructoras, algunas sofisticadas, otras tan simples como inutilizar el disco duro a martillazos. Sin ánimo de dar malas ideas se citan aquí algunos ejemplos procedentes de una lista de sabotajes que puede ser tan larga como lo que alcance la imaginación del sospechoso:

- Hacer un borrado seguro de todo el disco duro con herramientas especializadas o simplemente utilizando el comando “dd” desde una distribución Linux autoarrancable Live-CD.

- Utilizar herramientas de borrado seguro (*wiping*) para eliminar todo rastro de aquellos archivos que podrían resultar comprometedores, dejando el resto del disco duro intacto.
- Reinstalar el sistema operativo en un ordenador de sobremesa o portátil. Esta maniobra se combina a veces con un formateado del disco duro. Aunque no suele ser muy efectiva obliga al investigador a utilizar herramientas de tallado de archivos o *file carving* y a invertir una cantidad de tiempo considerable en el análisis del medio y la búsqueda de elementos de evidencia.
- Extraer el disco duro del ordenador y sustituirlo por otro limpio.
- Copiar archivos importantes desde un ordenador de sobremesa a una unidad de red, una llave USB o un ordenador portátil, borrando después la información original. Esta maniobra se puede complementar con el empleo de herramientas de borrado seguro para que resulte más efectiva.
- Sacar los archivos por impresora y borrarlos del disco duro.
- Cambiar la fecha y la hora del reloj con el objeto de crear elementos de evidencia provistos de marcas de tiempo incorrectas que corroboren una historia o una coartada.
- Autoenvío de correo electrónico para fabricar pruebas falsas.
- Cambiar las etiquetas de las cintas de *backup* para dificultar su localización o crear confusión a la hora de colocarlas en el robot, para que de este modo en el próximo *backup* quede sobreescrito el soporte en el cual presuntamente se hallan los elementos de evidencia buscados.
- Inutilizar la controladora del disco duro calentando los chips con un soplete de butano, o aplicando la corriente alterna de la red eléctrica directamente sobre los conectores de alimentación de la placa.
- Iniciar o programar un ciclo de desfragmentación del disco duro.
- ... y un largo y poco edificante etcétera.

En cuanto a los restantes soportes de datos –discos CD y DVD-ROM, llaves USB, disquetes, tarjetas de memoria, etc.–, más pequeños, ligeros y fáciles de transportar que un disco duro, no hace falta seguir añadiendo posibilidades. El propio lector podrá sacar consecuencias y considerarse prevenido al respecto.

### 11.2.2 Tecnologías antiforenses

Cuando el esfuerzo del delincuente, más que a la destrucción de elementos de evidencia va dirigido a una ocultación de los mismos mediante procedimientos sofisticados que presuponen un elevado nivel de conocimientos informáticos, entramos en un dominio nuevo, el de las técnicas antiforenses. Por tales se entienden aquellas manipulaciones dirigidas a dificultar la investigación forense a través de procedimientos que afectan a la existencia, cantidad o calidad de elementos de evidencia disponibles en la escena del delito. Las técnicas antiforenses también buscan entorpecer el funcionamiento de las herramientas, las operaciones de adquisición y el análisis de datos.

Más que impedir la investigación lo que se persigue mediante el empleo de técnicas antiforenses es conseguir que los analistas empleen en la resolución de un caso más tiempo del que se pueden permitir en función de los recursos disponibles y la carga de trabajo en los laboratorios de investigación forense. Para ello existen numerosos procedimientos que se sirven de las características de funcionamiento de los sistemas informáticos y de las propias herramientas forenses para generar falsos resultados y confundir al investigador. La intención del atacante en última instancia consiste en impedir que las alegaciones basadas en elementos de evidencia digital puedan ser utilizadas delante de los tribunales.

Como ejemplo de estrategias antiforenses podemos citar la ocultación de datos, el borrado de elementos de evidencia, la creación de pistas falsas y el ataque contra herramientas forenses utilizadas por el investigador. El atacante intenta lograr sus objetivos mediante el empleo de software de cifrado o aplicaciones que permitan el acceso a zonas del disco no alcanzables por el sistema operativo. Utilizar herramientas de encriptación o borrado seguro puede resultar eficaz dependiendo de las circunstancias. Probablemente sirva para quitar de en medio elementos de evidencia incriminadores, pero no ayuda al sospechoso a escapar de otros problemas evidentes por el mismo uso de técnicas antiforenses, como acusaciones por destrucción intencionada de pruebas y negativa a cooperar. Con frecuencia resulta más útil ocultar la información de modo que las herramientas del investigador forense pasen por encima de ella sin detectarla.

La utilidad Timestomp, perteneciente a una *suite* de herramientas para *hacking* ético denominada Metasploit, modifica los metadatos de archivos en particiones NTFS. Diversas utilidades de esteganografía pueden utilizarse para incluir un archivo de texto en otro de imagen o multimedia. Slacker, también de Metasploit, es capaz de dividir los datos en fragmentos y colocarlos dentro de los espacios existentes entre el final de los archivos y el comienzo del *cluster* consecutivo en particiones NTFS y FAT. La información también puede ir oculta en zonas reservadas del disco duro como la HPA (*Host Protected Area*) o la DCO (*Device Configuration Overlay*), de las que ya se trató en el capítulo 3. Estas áreas no son visibles para la BIOS ni para el sistema operativo, pero se puede acceder a ellas con la ayuda de herramientas especiales.

También existe la posibilidad de lanzar contra las herramientas forenses ataques basados en defectos de funcionamiento que pudieran haber sido detectados mediante el estudio de la documentación técnica sobre el software. La alteración de encabezados de los archivos por medio de Transmogrify (otra de las herramientas de la *suite* Metasploit) puede hacer que Foremost, Scalpel o cualquier otra utilidad de *data carving* confundan una imagen digital con un documento MS-Word. Del mismo modo, diversas anomalías en la programación de los módulos que implementan algoritmos para el cálculo de *hashes* podrían obstaculizar la defensa de pruebas electrónicas ante un tribunal.

## 11.3 DESAFÍOS PARA EL FUTURO

Una parte considerable de los retos que aguardan al informático forense en los próximos años no tienen que ver con los avances en la tecnología del hardware ni de los sistemas operativos, sino con un incremento exponencial en el volumen de datos y la extensión de esa nueva modalidad de organización distribuida de recursos informáticos y procesos de trabajo que se llama *cloud computing*.

### 11.3.1 Clusters

Es posible que de aquí a unos pocos años la arquitectura de las herramientas forenses actuales, basadas en plataformas del tipo EnCase, FTK o SMART comience a quedarse obsoleta. Las capacidades del software no guardan el paso con el incremento en la complejidad y el volumen de los datos que han de ser analizados en las investigaciones modernas. En un documento reciente sobre este tema (*A second generation computer forensic analysis system – Internet*: <http://www.dfrws.org/2009/proceedings/p34-ayers.pdf>) escrito por Daniel Ayers, consultor neozelandés de Informática Forense y propietario de la empresa Elementary Solutions Ltd., las herramientas forenses de primera generación, como él las denomina, consistentes en *suites* gráficas de fácil manejo y equipadas con

funcionalidades de todo tipo, desde calculadoras hexadecimales a *plugins* para el trazado de gráficas de tiempo, ya no resultan adecuadas para los fines de la investigación moderna.

La eficacia de este software está fuera de toda duda cuando se trata de analizar ordenadores de sobremesa o portátiles pertenecientes a un sospechoso, o cuando el objetivo consiste en hallar elementos de evidencia digital para documentar los cargos presentados contra pederastas o defraudadores particulares. Pero difícilmente podrían estar a la altura de lo que nos espera en un futuro próximo: ciberguerras, delincuencia organizada, computación en la nube, dispositivos móviles de gran potencia y cantidades de datos en el orden de los terabytes en montajes RAID y sistemas de almacenamiento distribuidos.

Aunque los desarrolladores invierten gran cantidad de recursos para ponerse al día y las nuevas versiones de los programas comerciales ya son capaces de trabajar en red, las perspectivas a medio y largo plazo se ven comprometidas por una serie de factores limitantes difícilmente superables: baja velocidad de proceso, inadecuada para el número y la capacidad de los soportes de datos (discos duros, *pendrives*, DVD, etc.) que cada vez en mayor cantidad colapsan las colas de trabajo de los laboratorios; cuellos de botella en los interfaces de entrada y salida, defectos de programación y la imposibilidad de llevar a cabo auditorías y revisiones inmediatas en el código de herramientas propietarias.

Así mismo y para las necesidades de investigación actuales y futuras, el software disponible en la actualidad presenta limitaciones en cuanto a planificación y control de tareas de análisis. Las capacidades de automatización se reducen a unos cuantos *scripts* y los conceptos de abstracción de datos son poco claros. Lo que el investigador forense busca no son archivos ni cadenas de caracteres, sino elementos de evidencia.

Una solución adecuada para paliar estas limitaciones de rendimiento podría consistir en utilizar arquitecturas Beowulf, un sistema compuesto por ordenadores de sobremesa convencionales con software libre –variantes de Linux y BSD– funcionando en paralelo unidos por interfaces Ethernet a través de *switches* de alto rendimiento. El autor del documento al que antes se hizo referencia menciona un prototipo Beowulf creado para fines experimentales: DELV (*Distributed Digital Forensics System*). DELV está compuesto por ocho nodos Linux, un servidor de archivos y una estación de control. Con este montaje se consigue aumentar notablemente la velocidad de operaciones típicas en Informática Forense, como búsqueda de cadenas de caracteres y procesamiento de expresiones regulares.

Estos incrementos se consiguen en gran parte gracias a las características típicas de funcionamiento de un *cluster* Beowulf. Los datos procesados en busca de

elementos de evidencia permanecen en todo momento en la RAM del sistema, la cual permite un acceso más rápido que los discos duros y otros sistemas de almacenamiento. En operaciones que impliquen el acceso normal a servidores de archivos o imágenes forenses, las mejoras en cuanto a rendimiento quedarían limitadas a un múltiplo equivalente al número de ordenadores que integran el *cluster*.

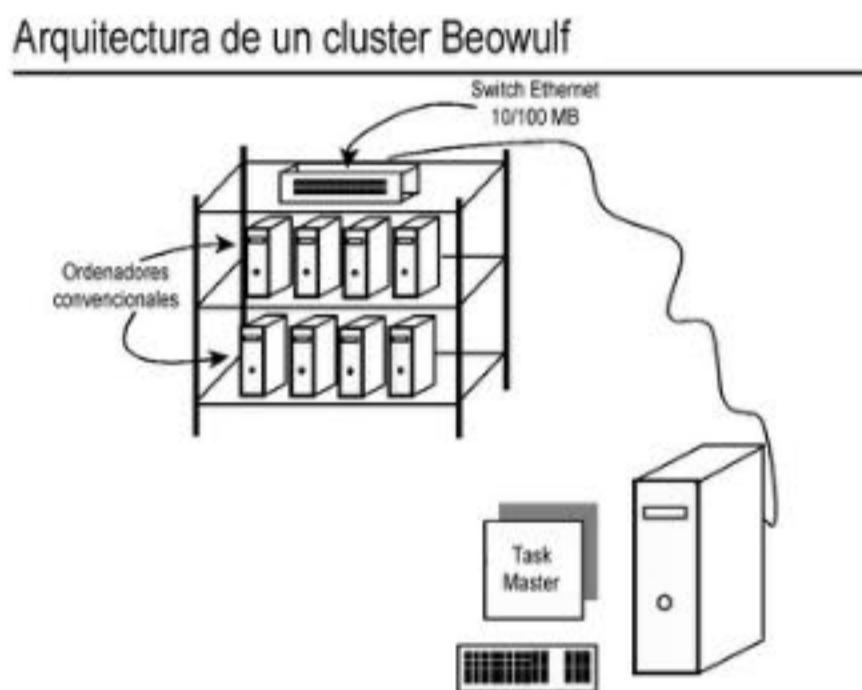


Figura 11.1. Esquema básico de un cluster Beowulf

### 11.3.2 Cloud computing

De un tiempo a esta parte el lector habrá leído en los medios noticias y artículos concernientes a esta modalidad nueva de computación que algunos consideran revolucionaria y con potencial suficiente para revolucionar el sector de las tecnologías de proceso de datos, mientras que otros ven en ella tan solo una moda pasajera como tantas otras que tuvieron su momento de gloria y pasaron a la historia sin dejar más que facturas de consultoría y presentaciones en PowerPoint. En eso que llaman la nube ya hay empresas que están sacando con éxito sus productos al mercado (con Google Apps, Amazon Web Services o Microsoft Azure como ejemplos más significativos). Aunque existen variados tipos de servicio las modalidades básicas consisten en el alquiler de máquinas virtuales –las AMI de Amazon– o de aplicaciones en modalidad S.a.a.S. (Software como Servicio) –Google–.

Conviene mencionar que mucho antes de que empezara a hablarse de la nube el lector probablemente ya utilizaba aplicaciones de *cloud computing*. Eso es lo que precisamente hace cada vez que escribe en su muro de Facebook, consulta el correo electrónico en Gmail, comparte un documento en Google Docs o sube sus

herramientas preferidas a un *dropbox* o un espacio de disco duro gratuito en Internet para tenerlas disponibles en cualquier lugar donde se encuentre.

Pese a los inconvenientes –pérdida de control de los datos, problemas de seguridad–, son varias las ventajas que ofrece la nube al usuario, como por ejemplo mayor flexibilidad a la hora de configurar la infraestructura de la empresa, así como una escalabilidad sin límites y considerables ahorros de costes. El investigador forense también se puede beneficiar de todo ello en cierta medida, pues para acceder a un ordenador, a no ser que se trate de una nube privada alojada en un servidor local, no es necesario desplazarse al lugar de los hechos. Basta introducir un identificador y una contraseña y el monitor mostrará en el acto la máquina virtual del sospechoso, con todos sus archivos y sus elementos de evidencia potenciales.

El inconveniente reside en que en la mayor parte de los casos será imposible acceder de forma directa a artefactos de interés forense como claves del Registro de Windows, archivos borrados, historiales de actividad del sistema y otros elementos por el estilo, a no ser –cosa por lo demás poco probable– que el proveedor de servicios en la nube admita desde el lado del usuario manipulaciones a bajo nivel para hacer posible el funcionamiento de aplicaciones forenses. A las dificultades de acceso hay que añadir la circunstancia de que todos los elementos de evidencia se pierden de manera irreversible cuando el usuario apaga o reinicia su máquina virtual.

En un entorno de investigación forense tradicional el especialista sigue una secuencia de pasos bien definidos: adquirir soportes de datos en el escenario del delito, aseguramiento de pruebas y preservación de la cadena de custodia. Posteriormente validación, análisis, interpretación y documentación del material con vistas a presentarlo delante de un tribunal. Todo este ciclo de actividad se desenvuelve en torno a unos medios probatorios (el hardware y los datos incautados) que en todo momento se encuentran bajo control, desde el momento de la incautación hasta la fecha del proceso. Ciertamente hay casos en los que las máquinas y sus discos duros tienen que quedarse en su sitio porque es imposible llevárselos sin paralizar un flujo de producción o causar perjuicios injustificables a la empresa. Pero, aun así, siempre se puede volver con una orden de registro a los locales de la organización o a las dependencias donde el proveedor de acceso aloja sus servidores.

En otras palabras, el problema fundamental con el *cloud computing* consiste en que la policía y los investigadores forenses no tienen acceso directo a la infraestructura física sobre la cual están implementadas las redes, los sistemas operativos y los soportes de datos. La situación se complica en aquellos casos en los que el gestor del servicio tiene su sede en un país determinado mientras que los

servidores de datos se encuentran en otro. De acuerdo con la mayor parte de las normativas, la intervención de un equipo sospechoso –suponiendo que pueda llevarse a cabo– tiene que hacerse de acuerdo con las leyes del estado en el que se encuentre ubicado el servidor. Pero en el mundo globalizado de las redes los emplazamientos no son físicos sino virtuales, y se hallan tan distribuidos como los propios recursos informáticos. ¿Qué hay, por ejemplo, de la posibilidad de que las máquinas virtuales o los sistemas sobre los cuales se ejecutan las aplicaciones de software se trasladen de unos países a otros según la estación del año o simplemente la hora del día, en busca de temperaturas más bajas –la energía para los sistemas de refrigeración es una de las principales partidas de gasto de los centros de datos– o tarifas eléctricas reducidas en el horario nocturno?

La solución de los problemas planteados por la nube requiere enfoques multidisciplinares que van más allá de la tecnología informática. Legisladores, responsables políticos, ingenieros, miembros de las fuerzas de seguridad, empresas y especialistas deberán colaborar con los expertos en la búsqueda de soluciones adecuadas. Una posibilidad podría consistir en que el mismo proveedor del servicio incorpore a su producto la capacidad de guardar informaciones evidenciales en previsión de que algún día llegue una orden de registro solicitándolas. Esto tampoco deja de suponer problemas, ya que la recuperación de datos en soportes sospechosos debe llevarse a cabo conforme a pautas estrictas, cuya menor vulneración –más que probable en caso de que los elementos de evidencia sean recolectados por personal sin capacitación forense– podría dar pie a la impugnación de la prueba delante de un tribunal.

En el momento actual lo único que se puede decir es que no existen métodos infalibles para la obtención de elementos de evidencia en entornos de *cloud computing*. En este sentido, la nube constituye uno más entre los numerosos ejemplos de aquellas situaciones, tan características del tiempo en que nos toca vivir, en las que el legislador, las autoridades públicas y en gran medida también los poderes políticos se ven desbordados por el avance vertiginoso y arrollador de la tecnología.

## 11.4 ALGUNAS RECOMENDACIONES

### 11.4.1 La vida no es bella...

Parecía fácil, ¿verdad? Acabábamos de descubrir el Santo Grial de la investigación forense, a saber, que según se desprende de los principios de la teoría de la información, establecidos hace más de medio siglo por el matemático norteamericano Claude Shannon, la mayor parte de energía no se emplea para escribir los datos, sino para borrarlos, y que este es el motivo por el cual los

diseñadores de sistemas operativos se conforman con que a la hora de eliminar un archivo los datos pertenecientes al mismo no sean eliminados, sino que el espacio que ocupaban quede marcado como disponible para grabar encima nuevos archivos. Supimos de la existencia de algunas herramientas comerciales y de código libre bastante útiles para frustrar los propósitos de ocultación del sospechoso. Aprendimos el manejo de algunas de ellas y nos quedamos fascinados cuando logramos recuperar nuestro primer archivo borrado. Algo que nadie podía ver, que ni siquiera sabíamos que estaba allí, pero que gracias a nuestro esfuerzo en el aprendizaje de la ciencia forense y a los inestimables servicios de TSK, vuelve milagrosamente a la vida desde su tumba digital en una llave USB o un disco duro externo recién laminado por su usuario.

Y de pronto, cuando ya teníamos las respuestas, van y nos cambian las preguntas. La Informática Forense, que durante más de diez años ha funcionado a base de procedimientos establecidos, validados y reconocidos por las autoridades judiciales de todos los países, ya no es lo que era ni volverá a serlo. La informática móvil, los entornos distribuidos, la complejidad de los casos de investigación y para colmo la nube pueden hacer que no solo los contenidos del presente libro, sino todo lo aprendido hasta la fecha, se quede desfasado en unos pocos años y requiera un constante esfuerzo de reciclaje.

Por lo tanto, si el lector pensaba que con unas cuantas nociones de sistemas de archivos y un software de recuperación de datos podía convertirse en investigador forense y ganarse la vida haciendo informes periciales, ya puede ir quitándose de la cabeza. No es tan sencillo. La Informática Forense del futuro no necesita funcionarios del teclado ni opositores versados en manuales de EnCase o Linux, sino gente perspicaz y con sentido común, entusiasta y bien entrenada en la aplicación del método científico. Alguien con talento para encontrar una aguja en un pajar antes que alguien experimentado en el manejo de información y el cumplimiento de trámites administrativos. En otras palabras: alguien con la mentalidad de un verdadero investigador. Este es, en resumen, el reto al que se enfrentan los investigadores en un futuro que ya es presente.

### 11.4.2 Para terminar

Aquí lo dejamos. Si quiere dedicarse a la investigación digital no es necesario que sea un fuera de serie de la informática, ni que lo sepa todo acerca de aplicaciones como TSK o Helix. Entender es más importante que saber, y con lo dicho no queda sino insistir en algunas cosas que le serán útiles para su trabajo. No divague en juicio ni haga afirmaciones caprichosas. Mantenga una actitud receptiva ante los avances de la tecnología. Esté al tanto de las últimas tendencias, lo mismo si se trata de desarrollos en su especialidad como de tipo general –telefonía móvil,

*cloud computing*, realidad aumentada, etc.–. Nunca se sabe cuándo el conocimiento puede llegar a sernos útil.

No olvide algo importante: en el mundo artificial todo está hecho por y para el hombre. Al igual que la maquinaria pesada y las instalaciones de la industria química no son en última instancia más que una proyección de los músculos y las vísceras del ser humano, los sistemas informáticos surgen, después de la red eléctrica y el teléfono, como extensiones del sistema nervioso. Detrás de cada artilugio técnico hay individuos de carne y hueso. Por este motivo, además de todos esos conocimientos técnicos, le conviene conocer al delincuente. Entender qué quiere y cuál es su *modus operandi*. Bien está que se sepa de memoria las especificaciones del sistema de archivos NTFS, o que domine con absoluta maestría la línea de comando de Linux. Pero tampoco debe perder de vista otras fuentes como la página web de la Guardia Civil, o las publicaciones de diversas consultoras e institutos de seguridad sobre temas como la organización de las cibermafias, perfiles de pederastas o la problemática del empleado desleal.

## BIBLIOGRAFÍA

---



---

ALTHEIDE, C.; CARVEY, H.; *Digital Forensics with Open Source Tools*, Syngress Elsevier, 2011.

AVIV, A. J.; GIBSON, K.; MOSSOP, E.; BLAZE, M. y S.; JONATHAN, M.; *Smudge Attacks on Smartphone Touch Screens*, Department of Computer and Information Science, University of Pennsylvania.

BARON, C.; *Adobe Photoshop Forensics. Sleuths, Truths and Fauxtography*, Thomson Course Technology, 2008.

BAUN, C.; KUNZE, M.; NIMIS, J.; TAI, S.; *Cloud Computing. Web-Based Dynamic IT Services*, Springer-Verlag, Berlin Heidelberg, 2011.

BUNTING, S.; WEI, W.; *The Official EnCase Certified Examiner Study Guide*, Sybex, 2006.

CARRIER, B.; *File System Forensic Analysis*, Addison Wesley, 2005

CARVEY, H.; *Windows Forensic Analysis DVD Toolkit*, Syngress, 2009.

CARVEY, H.; *Windows Forensics and Incident Recovery*, Addison Wesley Professional, 2004.

CASEY, E.; *Digital Evidence and Computer Crime*, Academic Press, 2004.

CASEY, E.; *Handbook of Digital Forensics and Investigation*, Academic Press, 2009.

CROWLEY; *CD and DVD Forensics. Handle, Examine and Process CD and DVD Evidence for Computer Forensics*, Syngress Publishing, 2007.

FARID, H.; *Digital Image Forensics*, Scientific American, June 2008

GARRIDO CABALLERO, J.; *Análisis Forense Digital en Entornos Windows*, Informática64, 2010,

GESCHONNECK, A.; *Computer-Forensik*, Dpunkt.Verlag GmbH, 2011.

GRUNDY, B. J.; *The Law Enforcement and Forensic Examiner's Introduction to Linux. A Practitioner's Guide to Linux as a Computer Forensic Platform*, LinuxLEO.com, 2008.

HOOG, A., *Android Forensics. Investigation, Analysis and Mobile Security for Google Android* Syngress, 2011.

HOOG, A.; STRZEMPKA, K.; *iPhone and iOS Forensics. Investigation, Analysis and Mobile Security for Apple iPhone, iPad and iOS Devices*, Syngress Elsevier, 2011.

JONES, K. J.; BEJTLICH, R.; ROSE, C. W.; *Real Digital Forensics: Computer Security and Incident Response*, Addison Wesley, 2005.

LASZEWSKI, T.; NAUDURI, P.; *Migrating to the Cloud. Oracle Client / Server Modernization*, Syngress Elsevier, 2012.

LESSARD, J.; KESSLER, G. C.; *Android Forensics: Simplifying Cell Phone Examinations*, Small Scale Digital Device Forensics Journal, vol. 4., September 2010.

RAMACHANDRAN, V.; *Backtrack 5 Wireless Penetration Testing. Master bleeding edge wireless testing techniques with BackTrack 5 – Beginner's Guide*, Packt Publishing, 2011.

RANKIN, K.; *Knoppix Hacks. Tips and Tools for Using the Linux Live CD to Hack, Repair and Enjoy your PC*, O'Reilly, 2008.

REIS, G.; *Photoshop CS3 for Forensics Professionals. A Complete Digital Imaging Course for Investigators*, Wiley, 2007.

ROSALES GARCÍA, M. A.; *Analisis forense en imágenes digitales*, Tesis de Maestría en Ciencias de Ingeniería y Microelectrónica. Instituto Politécnico Nacional, México D.F., 2008.

SCHROADER, A.; COHEN, T.; *Alternate Data Storage Forensics*, Syngress, 2007.

TROST, R.; *Practical Intrusion Analysis. Prevention and Detection for the Twenty-First Century*, Addison-Wesley, 2009.

WHITAKER, A.; NEWMAN, D.; *Penetration Testing and Network Defense. The practical guide to simulating, detecting and responding to network attacks*, Cisco Press, 2006.

## ÍNDICE ALFABÉTICO

### A

- ACPO ..... 246  
 ACPO Good Practice Guide for Computer-Based Electronic Evidence ..... 246  
 ADB ..... 239, 240, 241, 242  
 Adepto ..... 109, 163, 164, 165, 227, 275, 310  
 Adobe dng ..... 87  
 Adobe Photoshop Elements ..... 269, 271  
 Adquisición ..... 19, 28, 38, 45, 47, 48, 49, 51, 54, 55, 56, 64, 67, 72, 74, 102, 103, 104, 106, 107, 108, 109, 161, 162, 163, 164, 165, 166, 173, 219, 220, 226, 227, 228, 229, 232, 233, 235, 241, 245, 248, 249, 273, 274, 275, 278, 279, 283, 284, 285, 286, 287, 290, 292, 295, 296, 301, 308, 316, 318  
 ADSL ..... 177, 184  
 AFF ..... 71, 109  
 AIR ..... 53, 54, 56, 163, 295, 310  
 Amazon Web Services ..... 321  
 AMD ..... 308  
 Amenaza interna ..... 31  
 Análisis post mortem ..... 70, 96, 155  
 Analizador de tráfico ..... 189, 190  
 Android ..... 47, 145, 152, 209, 210, 211, 212, 213, 214, 215, 216, 219, 226, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 248, 251, 310, 328

### B

- Backtrack ..... 36, 50, 71, 195, 196, 203, 277, 295, 296, 297, 328  
 Backup ..... 38, 44, 219, 220, 221, 222, 223, 225, 281, 317  
 Bash ..... 19, 24, 73, 83, 90, 149, 154, 155, 160, 199  
 Bayer ..... 255, 256

### C

- API ..... 189, 213, 214, 233, 307  
 App Store ..... 213, 214, 217, 222, 228  
 Apple A4 ..... 212  
 Árbol (TID) ..... 189  
 Árbol de claves (registro de Windows) ..... 142  
 Archivo informático ..... 32, 82, 131, 266  
 Archivos abiertos ..... 96, 103, 141  
 Archivos borrados ..... 20, 38, 51, 67, 73, 78, 79, 81, 91, 108, 110, 113, 116, 123, 124, 129, 131, 137, 138, 146, 167, 214, 219, 220, 227, 228, 232, 241, 244, 295, 302, 310, 311, 322  
 ARM Cortex ..... 212  
 ARP ..... 185, 187, 193, 281, 283, 286  
 Arranque dual ..... 63, 75  
 Artefactos ..... 40, 125, 152, 168, 219, 257, 322  
 ASCII ..... 82, 85, 132  
 ASF/WMA ..... 89  
 Autopsy ..... 72, 73, 296, 310  
 AVI ..... 83, 88, 267, 268

### D

- BCC (Blind Carbon Copy) ..... 206  
 Beowulf ..... 320, 321  
 Bind ..... 199  
 Bios ..... 108, 292, 307, 319  
 Bit Locker ..... 282  
 BitTorrent ..... 89, 117  
 Blackberry ..... 209, 243  
 Blklks ..... 71  
 Bloqueador de escritura ..... 48, 64, 74, 107, 162, 245, 249  
 Bloqueo de pantalla ..... 216  
 Bluetooth ..... 210, 213, 217, 247  
 Botnet ..... 31, 36, 100  
 Bridge ..... 193  
 Buenas prácticas ..... 18, 25, 45, 196, 198, 233, 246, 261, 264, 265, 266, 313  
 Búsqueda de caracteres ..... 39, 111, 128, 129, 137, 178, 303
- Cabeceras del correo electrónico ..... 205  
 Caché ARP ..... 193  
 Cadena de custodia ..... 25, 37, 38, 42, 43, 45, 55, 56, 67, 106, 148, 164, 170, 173, 176, 245, 247, 265, 279, 285, 290, 313, 314, 322  
 Caine ..... 300, 301, 302, 310  
 Cámara digital ..... 25, 26, 63, 85, 219, 251, 252, 254, 255, 256, 265, 266, 267, 269  
 Capa de red ..... 179, 183, 185  
 Capa de transporte ..... 181, 183  
 Captain Nemo ..... 115, 310  
 Captura de memoria RAM ..... 104  
 Captura de paquetes con Wireshark ..... 194  
 CCD ..... 87, 254, 255, 266  
 Cellebrite ..... 226, 244  
 Chkrootkit ..... 168, 169, 170, 310  
 Chmod ..... 155, 241, 306  
 Ciberdelincuencia ..... 28  
 Ciberpunk ..... 31  
 Ciberterrorismo ..... 29  
 Cifrado ..... 39, 55, 137, 211, 287, 318  
 Claves ..... 141, 142, 144, 322  
 Claves (del registro de windows) ..... 143  
 CLF ..... 188  
 Clonación ..... 257  
 Cloud Computing ..... 19, 47, 235, 319, 321, 322, 323, 325  
 Cluster ..... 58, 64, 69, 78, 82, 319, 320, 321  
 Cluster slack ..... 69  
 CMOS ..... 254
- Dalvik ..... 235  
 Data carving ..... 92, 93, 114, 131, 138, 147, 167, 220, 243, 319  
 Data Expert Witness ..... 107  
 Data runs ..... 57, 137  
 Date ..... 97, 158  
 DBX folders ..... 127  
 DC3dd ..... 52, 53, 310  
 DCO ..... 38, 64, 319  
 DD ..... 38, 50, 51, 52, 53, 56, 73, 74, 76, 77, 79, 80, 81, 92, 104, 106, 109, 110, 111, 116, 153, 162, 163, 164, 165, 166, 167, 168, 196, 227, 231, 232, 235, 241, 242, 243, 245, 275, 287, 295, 296, 300, 310, 316

DDFLDD ..... 52  
 DDRescue ..... 52, 296, 300, 310  
 Debian ..... 291  
 Delito informático ..... 27, 28, 30  
 DELV ..... 320  
 DFRWS ..... 18  
 DHCP ..... 174, 175, 185  
 Dirección MAC ..... 84, 175, 187, 196, 280  
 Directorio ..... 58, 61, 65, 68, 70, 72, 74, 78, 80, 81, 83, 84, 93, 107, 118, 119, 121, 123, 124, 137, 149, 152, 153, 154, 155, 156, 157, 158, 167, 169, 170, 238, 239, 241, 242, 292, 304  
 Disco IDE ..... 48, 51  
 Disk investigator ..... 129, 130, 310  
 Diskexplorer ..... 136, 137, 138, 311  
 Dispositivos móviles ..... 20, 47, 62, 145, 152, 209, 210, 212, 213, 214, 216, 220, 221, 226, 227, 233, 234, 245, 246, 248, 249, 250, 320  
 DMESG ..... 306  
 D-Messagebus ..... 149, 150, 304, 306  
 DMZ ..... 185  
 Documento electrónico ..... 262  
 Documentos ..... 24, 29, 32, 55, 60, 65, 67, 70, 81, 82, 84, 85, 86, 91, 95, 110, 113, 114, 117, 119, 129, 131, 132, 133, 134, 135, 136, 153, 154, 156, 164, 177, 182, 192, 216, 219, 234, 236, 241, 243, 253, 262, 268, 288, 299, 315  
 Documentos rtf ..... 85  
 Drm ..... 89

**E**

E.l.a. ..... 260, 261  
 Easyrecovery ..... 139, 310  
 Editor hexadecimal ..... 68, 70, 82, 83, 130, 132  
 Eindeutig ..... 128  
 Elementos de evidencia ..... 38, 39, 40, 44, 45, 52, 64, 65, 79, 82, 95, 96, 104, 107, 108, 110, 111, 112, 114, 129, 132, 134, 135, 141, 144, 174, 175, 176, 178, 191, 198, 219, 220, 222, 226, 228, 229, 244, 247, 262, 267, 271, 273, 275, 278, 281, 283, 284, 286, 288, 292, 293, 302, 313, 317, 318, 320, 321, 322, 323  
 EMACS ..... 304  
 Emule ..... 89, 117  
 Encase ..... 19, 39, 49, 50, 54, 64, 70, 71, 81, 91, 92, 106, 107, 108, 110, 111, 112, 113, 114, 115, 116, 138, 145, 146, 161, 167,

220, 226, 232, 287, 295, 306, 310, 319, 324, 327  
 ENCE (*EnCase Certified Examiner*) ..... 110  
 Enumeración ..... 35  
 EOF ..... 78  
 Espacio no asignado ..... 38, 73, 114, 128, 178, 227, 241  
 Ethernet ..... 175, 179, 185, 186, 187, 190, 192, 194, 196, 297, 320  
 Etiquetas XMP ..... 86  
 Eudora ..... 126, 128  
 Evidor ..... 129, 130, 310  
 EWF ..... 50, 71  
 Excel ..... 84, 91, 119, 124, 126, 131, 218, 306  
 EXIF ..... 86, 93, 135, 136, 217, 218, 266, 267, 270, 271  
 EXIFtool ..... 135, 136, 148, 268, 269, 310  
 Exploits ..... 35  
 Expresiones regulares ..... 114, 129, 320  
 Ext2 ..... 59, 60, 68, 116, 140, 151, 166, 167, 275  
 Ext3 ..... 37, 59, 60, 68, 148, 150, 151, 275  
 Ext4 ..... 37, 60, 151

**F**

Facebook ..... 219, 321  
 Fat ..... 58, 59, 60, 61, 62, 65, 67, 68, 76, 77, 78, 79, 95, 116, 136, 137, 138, 139, 140, 151, 311, 319  
 Fat16 ..... 59, 75, 107, 151  
 Fat32 ..... 59, 75, 77, 78, 151, 289, 292  
 FBI ..... 197  
 Fdisk ..... 64, 76, 167  
 Fecha y hora del sistema ..... 97  
 Fedora ..... 149, 269  
 FHS ..... 152, 161  
 File slack ..... 69, 111  
 Filedisk ..... 116, 310  
 Filtros ..... 191, 195, 197, 256  
 Firefox ..... 118, 119, 120, 121, 122, 125, 128  
 Firewall ..... 174, 290  
 Firewire ..... 48, 50, 108  
 Firma característica ..... 82  
 Firma digital ..... 176  
 First responder ..... 25  
 FLS ..... 71, 72, 79, 80, 166, 167  
 Foca ..... 134, 135, 310  
 Footprinting ..... 33  
 Foremost ..... 92, 93  
 Fping ..... 33, 202, 203  
 Fport ..... 99

Free software foundation ..... 146  
 Froyo ..... 233  
 Fsstat ..... 71, 76, 77, 79  
 Ftk ..... 19, 39, 54, 64, 70, 81, 91, 92, 106, 108, 109, 112, 113, 114, 115, 138, 145, 146, 147, 167, 232, 275, 295, 301, 306, 310, 319  
 Ftk imager ..... 108  
 FTP ..... 34, 180, 276

**G**

Geolocalización ..... 211, 216  
 Get ..... 197  
 Getdataback ..... 116, 138, 139, 295, 311  
 Gif ..... 86  
 Gimp ..... 253, 254, 269, 270, 271, 298, 311  
 Gingerbread ..... 233  
 Giróscopo ..... 209, 213  
 Gnome ..... 149, 150, 275, 292, 304  
 GNU GPL ..... 151  
 Google ..... 33, 34, 83, 122, 126, 188, 211, 213, 214, 217, 234, 310, 321, 328  
 Google Chrome ..... 122, 188  
 GPG (cifrado de archivos) ..... 137  
 GPS ..... 85, 135, 136, 209, 210, 211, 213, 217, 218, 267  
 Guidance Software ..... 49, 70, 110

**H**

Hacker ..... 31, 245  
 Hacking ético ..... 134, 296, 319  
 Hal ..... 149, 150, 304, 306  
 Hash ..... 39, 49, 51, 55, 56, 67, 91, 108, 149, 150, 163, 164, 265, 276, 289, 290, 293, 298  
 Hashkeeper ..... 91  
 Helix ..... 109, 163, 165, 273, 274, 275, 276, 277, 278, 279, 281, 282, 284, 288, 289, 291, 292, 293, 294, 301, 311, 324  
 Helix pro receiver ..... 285  
 Helix3 pro™ ..... 274, 276, 277, 279, 285, 287, 288, 289, 290, 292  
 HFS ..... 20, 37, 59, 61, 62, 140, 151, 213, 231  
 HFS+ ..... 20, 37, 59, 61, 62, 140, 151, 213, 231  
 Hibernación ..... 106  
 Historial de Internet ..... 117, 118, 119, 120, 121, 165, 241, 302, 311, 312  
 Hotspots ..... 218  
 HPA ..... 38, 64, 72, 319  
 HTC ..... 211, 213, 214, 241

**I**

IANA ..... 182, 206  
 ICANN ..... 183, 184  
 ICAT ..... 71, 81, 168  
 Ice Cream ..... 233  
 Icmp ..... 34, 202, 203, 204  
 Iehist ..... 120, 311  
 ILS ..... 71, 72, 167  
 Imagen a bajo nivel ..... 38, 45, 47, 64, 164, 166, 220, 227, 229, 235, 241, 275, 284  
 Imagen digital ..... 253, 256, 257, 260, 261, 262, 263, 319  
 IMAP ..... 126  
 IMEI ..... 216  
 Index.dat ..... 119, 311  
 Indexado ..... 113  
 Info2 ..... 123, 124  
 Información volátil ..... 96, 97, 102, 158, 159, 161, 176, 274, 277, 278, 281, 285, 287  
 Ingeniería social ..... 18, 100  
 Inode ..... 60, 61, 74, 79, 81, 157, 168  
 Internet ..... 17, 18, 19, 23, 24, 29, 30, 31, 33, 34, 35, 36, 40, 44, 63, 67, 86, 88, 91, 99, 111, 112, 117, 118, 119, 120, 121, 122, 125, 126, 145, 151, 154, 177, 179, 180, 182, 183, 184, 185, 188, 197, 198, 199, 200, 202, 203, 204, 205, 206, 207, 211, 212, 215, 217, 218, 222, 228, 229, 234, 237, 242, 244, 252, 259, 260, 274, 276, 297, 306, 312, 315, 316, 319, 322  
 Internet explorer ..... 99, 111, 118, 119, 120, 122, 125, 188, 312  
 Interpolación ..... 256, 257  
 Intranet ..... 192  
 IOS ..... 209, 211, 212, 216, 221, 222, 225, 228, 231, 233, 241, 328  
 IPAD ..... 62, 210, 221, 222, 225, 233, 328  
 iPhone ..... 20, 62, 152, 196, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 236, 240, 244, 245, 247, 311, 328  
 iPhone 4 ..... 212  
 Iphone backup extractor ..... 224

iPod ..... 62, 221, 232, 302  
 Iptc-iim ..... 267  
 IPV6 ..... 101, 183  
 ISO ..... 62, 116, 267, 270, 275, 276, 292, 293, 296, 299  
 Itouch ..... 62, 221, 222, 233  
 Itunes ..... 89, 213, 219, 220, 221, 222, 223, 224, 226, 233, 236, 237, 302, 311

**J**

Jailbreaking ..... 214, 220, 228, 229, 230, 231, 241  
 Jaula de faraday ..... 216, 219, 247  
 Javascript ..... 84  
 Jerarquía unificada de directorios ..... 152  
 Journaling ..... 37, 57, 59, 60, 62, 66, 72, 110, 148, 150, 293, 298, 300, 316  
 Jpeg ..... 65, 86, 218, 260, 265, 266, 267, 270

**K**

Kde ..... 150, 292, 298, 304  
 Kernel ..... 90, 104, 151, 152, 159, 169, 214, 233, 242, 300, 304, 307, 308  
 Kff (known file filter) ..... 113  
 Knoppix ..... 71, 110, 150, 203, 274, 276, 277, 291, 292, 297, 298, 299, 311, 328  
 Kodak ..... 136, 255, 268

**L**

LAN ..... 184, 192, 280  
 Ley de enjuiciamiento civil ..... 262  
 Ley de firma electrónica ..... 29  
 Ley de propiedad intelectual ..... 29  
 Ley de protección de datos ..... 29, 198, 315  
 Ley de servicios para la sociedad de la información ..... 29  
 Ley general de telecomunicaciones ..... 29  
 Línea de tiempo ..... 33, 40, 72, 96, 165, 166  
 Linen ..... 49, 50, 295, 311  
 Linkedin ..... 219  
 Linux ..... 19, 24, 36, 37, 49, 50, 51, 52, 56, 59, 60, 61, 62, 63, 64, 65, 66, 68, 70, 71, 72, 73, 74, 75, 76, 81, 82, 83, 85, 88, 90, 92, 95, 104, 109, 110, 119, 120, 124, 137, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 157, 159, 160, 161, 162, 165, 166, 167, 169, 176, 177, 189, 190, 192, 193, 194, 195, 196, 200, 203, 209

214, 220, 224, 230, 231, 233, 234, 238, 239, 240, 241, 242, 267, 268, 269, 273, 274, 275, 276, 277, 279, 281, 284, 285, 289, 291, 295, 297, 298, 299, 300, 301, 302, 304, 306, 307, 308, 309, 310, 311, 312, 316, 320, 324, 325, 328  
 Live-CD ..... 24, 177, 273, 274, 276, 291, 297, 299, 311  
 Llave USB ..... 24, 32, 59, 62, 73, 75, 76, 91, 93, 103, 141, 143, 159, 177, 227, 274, 285, 292, 299, 317, 324  
 Lotus Notes ..... 133  
 Luks ..... 282

**M**

M4P ..... 89  
 M4R ..... 89  
 Mactime ..... 72, 167  
 Mac-time ..... 157  
 Magic ..... 83, 168  
 Mainframe ..... 306  
 Manifest.plist ..... 225  
 Maps ..... 216, 217  
 Máquina virtual ..... 89, 214, 235, 293, 306, 307, 308, 309, 322  
 Máquina virtual Dalvik ..... 214  
 Marcas de tiempo ..... 40, 57, 58, 65, 82, 86, 93, 157, 166, 167, 176, 301, 317  
 MBR ..... 51, 79, 80, 167  
 MD5 ..... 39, 49, 53, 55, 56, 91, 108, 112, 116, 163, 265, 311  
 MD5Deep ..... 170, 311  
 MD5Sum ..... 56, 196  
 Memoria flash ..... 62, 63, 65, 212, 220, 242  
 Metadata assistant ..... 133, 134, 311  
 Metadatos ..... 40, 60, 64, 65, 66, 68, 71, 72, 81, 82, 84, 85, 86, 87, 88, 89, 93, 131, 132, 133, 134, 135, 136, 148, 157, 167, 217, 218, 253, 266, 267, 268, 271, 302, 310, 311, 319  
 Metasploit ..... 36, 319  
 MFT ..... 57, 58, 61, 65, 68, 79, 82, 136, 137, 138  
 Microsoft Azure ..... 321  
 MKV ..... 88  
 MMLS ..... 71, 74, 75, 76, 167  
 Mobileme ..... 216  
 Modelo de capas ..... 34, 62, 65, 66, 71, 81, 180  
 Modo avión ..... 247  
 Montaje ..... 24, 67, 108, 116, 148, 149, 150, 152, 153, 159, 164, 254, 255, 259, 274,

275, 279, 282, 292, 300, 302, 303, 304, 306, 316, 320  
 Mount Image Pro ..... 116, 119, 124, 311  
 MP3 ..... 26, 47, 59, 89, 117, 136, 218, 268  
 Mpeg-1 ..... 83, 87  
 Mpeg-2 ..... 87  
 Mpeg-3 ..... 89  
 Mpeg-4 ..... 87  
 MSDOS ..... 19, 58, 59, 69, 70, 76, 81, 95, 107, 167, 304, 308  
 MS-Office ..... 65, 84, 90, 95, 131, 136  
 MTA ..... 205, 206  
 Multimedia ..... 60, 87, 88, 92, 215, 218, 268, 319  
 Multiplexado (MID) ..... 189  
 MySpace ..... 219

**N**

NetBios ..... 99, 101, 163, 164, 181, 189  
 NetCat .. 49, 98, 104, 160, 161, 231, 232, 285, 288, 296  
 Netflow ..... 174, 185  
 Netstat ..... 97, 98, 99, 101, 159, 160, 176, 181  
 Nist ..... 91  
 Nombres de dominio ..... 159, 200, 203  
 Notes app ..... 217  
 Notmyfault ..... 105  
 Nslookup ..... 199  
 NTFS ..... 37, 57, 58, 59, 60, 61, 65, 67, 68, 76, 79, 95, 116, 136, 137, 138, 139, 140, 148, 149, 151, 275, 300, 311, 319, 325  
 Ntuser.dat ..... 142  
 Nube ..... 47, 306, 320, 321, 322, 323, 324

**O**

Office 2007 ..... 133  
 Offset ..... 72, 77  
 Ole ..... 131  
 Ole (objetos incrustados) ..... 127  
 Open handset alliance ..... 214, 234, 237  
 Open Office ..... 84, 85, 119, 128  
 Open XML ..... 84  
 Opendocument ..... 85  
 OpenGL ..... 214  
 Opera ..... 118, 119, 125  
 Oracle ..... 308, 309, 328  
 Orden de volatilidad ..... 283  
 OS2/Warp ..... 308  
 OSI ..... 34, 62, 180, 186

OSX ..... 20, 37, 50, 62, 71, 81, 85, 86, 90, 95, 120, 145, 152, 162, 193, 209, 212, 213, 222, 224, 229, 230, 231, 233, 238, 273, 274, 275, 276, 277, 279, 281, 284, 285, 289, 308, 310, 311  
 Outlook ..... 24, 126, 127, 128, 133, 206  
 Oxygen Forensics Suite ..... 226

**P**

Palm pilot ..... 210, 226  
 Papelera de reciclaje ..... 122  
 Papelera de reciclaje windows ..... 122  
 Paraben device seizure ..... 226  
 Paraben's e-mail examiner ..... 128, 311  
 Partición ..... 37, 51, 56, 58, 59, 60, 61, 63, 64, 66, 67, 68, 69, 75, 76, 78, 79, 106, 107, 136, 137, 138, 140, 149, 151, 152, 154, 167, 222, 228, 229, 231, 232, 241, 242, 244, 282, 292, 293, 297, 298, 300  
 Partición extendida ..... 76  
 Password breaker ..... 225  
 Password breaker (iPhone) ..... 225  
 Pata ..... 149, 150, 162  
 PDF ..... 65, 79, 80, 81, 84, 85, 91, 131, 133, 135, 156, 218, 268, 285, 290  
 Perl ..... 19, 90, 144, 148, 269  
 Permisos ..... 57, 59, 60, 82, 136, 154, 155, 156, 219, 222, 235, 236, 239, 241, 292, 297, 304, 305  
 Permisos de ejecución ..... 82, 241, 304, 306  
 Pgp-bootguard ..... 282  
 Pgp-pgpguard ..... 282  
 Phishing ..... 23, 30  
 PID ..... 103, 158, 159, 189  
 Pila de protocolos ..... 62, 179, 185, 202, 212  
 Ping ..... 33, 34, 199, 202, 203  
 Png ..... 65, 86, 135, 255, 267  
 Pop3 ..... 126, 210, 250  
 Pornografía infantil ..... 21, 24, 29, 36, 44, 100, 315  
 Post ..... 197  
 Privacidad ..... 126, 210, 250  
 Privilegios ..... 35, 36, 52, 73, 76, 149, 154, 155, 156, 162, 168, 169, 170, 228, 230, 236, 239, 241, 242, 243, 245, 278, 293  
 Protocolo ..... 34, 99, 125, 179, 180, 183, 186, 187, 188, 189, 190, 195, 196, 247, 249  
 Prueba digital ..... 44, 316  
 Prueba personal ..... 44  
 Pruebas de penetración ..... 134

Plist ..... 102  
 Psloggedon ..... 100  
 Pst ..... 127  
 Pstools ..... 100, 103, 104, 105, 311  
 Pst-viewer ..... 127  
 Puente ..... 193  
 Puerta de enlace por defecto ..... 179, 183, 184  
 Puerta trasera ..... 98, 165, 177  
 Puertos ..... 34, 35, 96, 98, 99, 100, 158, 160, 181, 182, 192, 193, 197, 239, 240  
 Puertos dinámicos ..... 182  
 Punto de montaje ..... 149, 152  
 Python ..... 19, 90, 148

**Q**

Quarkxpress ..... 85  
 Quicktime ..... 88, 218, 268

**R**

Raid ..... 56, 287, 320  
 Ram/nand ..... 227  
 Raw ..... 51, 86, 87, 108, 255, 265, 266, 287  
 Realidad aumentada ..... 325  
 Received ..... 205, 206  
 Red hat ..... 291  
 Red pública ..... 183, 195  
 Red virtual ..... 174, 308  
 Redirecciónamiento ..... 81, 123, 124, 240  
 Reg (extensión del registro de Windows) ..... 143  
 Registro de Windows ..... 141  
 Registry Hives ..... 142  
 Registry Recovery ..... 143, 144, 312  
 Regripper ..... 144  
 Reiserfs ..... 37, 61, 62, 148, 150, 275  
 Retoque fotográfico ..... 83, 85, 86, 95, 135, 253, 254, 256, 257, 258, 261, 266, 268, 269, 298, 299, 310  
 Return-path ..... 205, 206  
 Rifiuti ..... 124, 311  
 Rkhunter ..... 168, 169, 170, 311  
 Rooting ..... 236, 240, 241, 242  
 Rootkit ..... 36, 143, 160, 165, 168, 169, 177, 178, 277  
 Router ..... 32, 175, 176, 179, 183, 184, 185, 192, 197, 204  
 R-studio ..... 140, 311  
 RTF ..... 85, 133  
 Runtime Android ..... 235  
 Russian Business Network ..... 197

**S**

S.A.A.S. (software como servicio) ..... 321  
 Safari ..... 118, 216, 218  
 Safeboot ..... 282  
 Samba ..... 163, 189  
 Samsung Galaxy ..... 211, 212, 215, 235, 237, 238, 242, 244, 248  
 Samsung Kies ..... 219, 236, 237  
 SATA ..... 48, 49, 51, 54, 107, 150, 162, 163  
 Script kiddie ..... 31  
 Scripts de arranque ..... 150, 274, 304  
 SCSI ..... 48, 53, 163  
 SDK (Software Development Kit) ..... 237  
 Sector Slack ..... 69  
 Sectores ..... 51, 52, 57, 63, 65, 68, 69, 70, 75, 77, 8, 79, 111, 114, 129, 131, 137, 167, 227  
 Sectorspy ..... 129, 311  
 Servidor DNS ..... 179  
 Servidor web ..... 125, 250  
 Setgid ..... 156  
 Setuid ..... 156  
 SGL ..... 214  
 Sha ..... 39, 55, 91, 112  
 Shodan ..... 33, 34  
 Shutdown ..... 153, 161  
 Silbido de la muerte ..... 202  
 Sincronización ..... 219, 220, 221, 222, 223, 224, 225, 236, 237, 238, 241, 311  
 Sistema de archivos ..... 55, 56, 57, 58, 60, 61, 62, 64, 65, 66, 68, 69, 74, 75, 76, 80, 81, 82, 92, 93, 95, 111, 112, 114, 129, 136, 137, 138, 140, 147, 149, 155, 162, 167, 168, 213, 227, 236, 241, 242, 243, 267, 282, 284, 325  
 Sistema de detección de intrusos ..... 174  
 Slack de archivo ..... 69  
 Slack de disco duro ..... 69  
 Slack RAM ..... 69  
 Slackware ..... 150, 291, 302, 303, 304, 305, 306, 312  
 Sleuth kit ..... 19, 64, 70, 71, 110, 138, 310  
 Smart ..... 19, 81, 108, 312, 319  
 Smartphone ..... 135, 210, 211, 213, 215, 216, 218, 219, 236, 237, 245, 251  
 SMB ..... 187, 189  
 SMS ..... 211, 215, 217, 218, 237  
 SMTP ..... 34, 205  
 Smudge attack ..... 252  
 Sniffer ..... 176, 190

Soporte de datos ..... 25, 37, 38, 39, 45, 47, 49, 50, 51, 52, 55, 56, 62, 65, 67, 68, 70, 72, 73, 75, 78, 79, 81, 91, 110, 112, 113, 128, 130, 139, 143, 152, 160, 233, 235, 266, 287, 298, 301

Spam ..... 30, 100, 125, 205, 207  
 Sqlite ..... 72, 120, 122, 211, 212, 214, 216, 218, 220, 224, 240, 312

Sqliteman ..... 122

SSH ..... 230, 231

SSL ..... 214

Stuxnet ..... 316

Superusuari ..... 52, 76, 136, 153, 154, 155, 162, 236, 239, 241, 242, 243, 297

Swap ..... 64, 76, 159

Swgit ..... 263, 264

Switch ..... 192, 193, 198

Sysinternals ..... 100, 105, 119, 301, 312

Syslog ..... 176

Systemdump ..... 105

Systemrescued ..... 71, 110, 150, 203, 277, 299, 300, 312

**U**

Ubuntu ..... 51, 71, 73, 74, 92, 147, 149, 150, 162, 193, 203, 268, 274, 275, 277, 291, 292, 293, 295, 297, 309

UDEV ..... 149, 150, 239, 304

UDP ..... 34, 96, 97, 98, 99, 160, 181, 204

UFED ..... 226, 244, 245

Unicode ..... 129, 132

Unix ..... 19, 50, 60, 62, 65, 70, 71, 79, 81, 82, 104, 110, 145, 147, 150, 152, 153, 157, 161, 162, 163, 166, 169, 189, 229, 230, 231, 281, 303, 304, 308, 310

USB debugging ..... 238

Usuario (UID) ..... 189

**V**

Video digital ..... 87

Virtualbox ..... 295, 308, 309, 312

Visualbasic ..... 65

Vmlinuz ..... 152

Vmware ..... 293, 295, 307, 308

Vmware player ..... 307

Vmware workstation ..... 307

Volcado de memoria ..... 44, 105

Volumen ..... 63, 64, 76, 78, 86, 91, 111, 113, 154, 164, 176, 179, 197, 198, 226, 232, 286, 292, 319

**W**

Wan ..... 179, 180, 184

Wap ..... 185

Warez ..... 36, 100, 196, 315

Wav ..... 88, 267, 268

Whois ..... 33, 199, 200, 201, 202, 203, 205, 312

Wifi ..... 175, 195, 196, 213, 216, 217, 296, 297

Win32dd ..... 104

Windows ..... 19, 23, 36, 37, 50, 57, 58, 59, 60, 63, 64, 65, 66, 67, 68, 70, 71, 74, 75, 76, 79, 81, 82, 83, 85, 88, 89, 90, 91, 95, 96, 98, 100, 102, 104, 105, 106, 108, 109, 110, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 127, 128, 129, 132, 135, 136, 137, 141, 142, 143, 144, 145, 147, 149, 151, 152, 154, 157, 160, 161, 176, 186, 188, 190, 192, 193, 194, 196, 203, 209, 210, 220, 222, 223, 224, 233, 238, 243, 267, 268, 271, 273, 274, 275, 276, 277, 279, 281, 284, 285, 289, 290, 297,

298, 299, 300, 301, 302, 304, 306, 307,  
308, 309, 310, 311, 312, 322, 327  
Windows 7 ..... 119, 142, 277  
Windows mobile ..... 209  
Windows phone ..... 243  
Winhex ..... 130, 131, 132  
Wintaylor ..... 301, 302  
Wireshark ..... 176, 190, 191, 192, 193, 194,  
195, 196, 197, 198, 199, 297, 312  
WMV ..... 88  
Wordpad ..... 85, 128  
Write blocker ..... 38

**X**

XML ..... 65, 84, 85, 133, 135, 268

XMP ..... 86, 88, 136, 267, 268, 271  
X-ways forensics ..... 130, 131, 312  
X-ways trace ..... 119, 120, 312

**Y**

YouTube ..... 217

**Z**

Zip ..... 85, 133, 153, 268  
Zona desmilitarizada ..... 185

SYNAPS PREMIUM

# Introducción a la Informática Forense

Actualmente las tecnologías de la información constituyen un elemento indispensable para el funcionamiento de organizaciones y empresas de todo tipo. La ubicuidad de medios informáticos, combinada con el crecimiento imparable de Internet y las redes durante los últimos años, abre un escenario de oportunidades para actos ilícitos (fraude, espionaje empresarial, sabotaje, robo de datos, intrusiones no autorizadas en redes y sistemas y un largo etcétera) a los que es preciso hacer frente entendiendo las mismas tecnologías de las que se sirven los delincuentes informáticos, con el objeto de salirles al encuentro en el mismo campo de batalla. Parte vital en el combate contra el crimen es una investigación de medios digitales basada en métodos profesionales y buenas prácticas al efecto de que los elementos de evidencia obtenidos mediante la misma puedan ser puestos a disposición de los tribunales. Se debe hacer con las suficientes garantías en lo tocante al mantenimiento de la cadena de custodia y al cumplimiento de aspectos esenciales para el orden legal del estado de derecho, como el respeto a las leyes sobre privacidad y protección de datos y otras normativas de relevancia similar.

La Informática Forense es la disciplina que se encarga de la adquisición, el análisis y la valoración de elementos de evidencia digital hallados en ordenadores, soportes de datos e infraestructuras de red, y que pudieran aportar luz en el esclarecimiento de actividades ilegales perpetradas en relación con instalaciones de proceso de datos, independientemente de que dichas instalaciones sean el objetivo de la actividad criminal o medios utilizados para cometerla. El propósito de esta obra consiste en introducir al lector, de manera resumida y clara, en los principios, métodos, las técnicas fundamentales y las implicaciones jurídicas de la investigación informática forense. A tal efecto se dará a conocer, con sencillez y mediante un número de ejemplos, cómo sacar partido a las soluciones, tanto propietarias como de código libre, utilizadas en la actualidad por los profesionales de la investigación forense. He aquí, entre otros, algunos de los temas tratados:

- Principios y metodología de la investigación de soportes de datos.
- Investigación forense de sistemas Microsoft Windows.
- Investigación forense de sistemas Linux/Unix.
- Investigación forense de dispositivos móviles.
- Investigación en redes informáticas e Internet.
- Investigación de imágenes digitales.
- Herramientas de software y distribuciones Linux para la investigación forense.



Ra-Ma®

SYNAPS PREMIUM