浏览器取证技术①

陶姿邑[1], 毕善为[2]

[1](陕西中医学院, 西安 712046)
[2](日立电梯(中国)陕西分公司, 西安 712046)

摘 要: 随着信息时代的来临, 一些不法分子在实施犯罪之前往往会上网查询信息, 他们所用的浏览器便成了司法机关取证的关键. 能否提取有效的犯罪线索或证据, 取决于浏览器取证方法的好坏, 本文介绍了目前主流的火狐浏览器、IE 浏览器的取证技术, 概述了 IE 缓存文件和基于 SQLite 数据库的火狐浏览器历史系统的日志文件结构, 提出了信息恢复方法. 通过对已删除日志文件或缓存文件信息提取, 来达到获取证据的目的, 分析用户的行为.

关键词: 浏览器取证; SQLite 数据库; 日志文件; 信息提取

Overview of Browser Forensics Technology

TAO Zi-Yi[1], BI Shan-Wei[2]

[1](Shanxi University of Chinese Medicine, Xi'an 712046, China)
[2](Hitachi Elevator(China), Shanxi Branch, Xi'an 712046, China)

Abstract: With the advent of the information age, some criminals always tend to query information from the Internet before they engaged in criminal activity. So the browser they used has become the key to the forensics of judicial authorities. Whether we can extract the effective evidence of crime depends on the forensics method of browser. This article introduces the forensics technology of Firefox and IE browser which are the current mainstream browsers, outlined the browser temporary file structure, such as the IE cache file and the SQLite database log files of the Firefox, proposed information recovery method. It can collect evidence and analyze the user's behavior by extract the information of the deleted log files or cache files.

Key words: browser forensic; SQLite database; log file; information extraction

1 引言

随着信息技术和网络的不断发展,利用互联网来查询犯罪信息或以网络计算机为目标的犯罪动越来越多, 对人民的合法权益造成了破坏. 如何最大限度地获取网络犯罪相关的历史信息证据,如何恢复犯罪分子已经删除的历史信息提取相关证据,将犯罪分子绳之以法,已成为取证工作者和计算机科学领域中急需解决的问题. 浏览器取证是打击计算机犯罪的有力工具及手段,为了提高打击计算机犯罪的能力,需要对浏览器取证领域进行深入的研究,不但需要开发切实有效的取证方法,也需要对浏览器取证领域的取证定义、取证标准、取证程序等进行研究.

2 相关工作

目前大多数的浏览器取证研究主要针对于特定的浏览器或是分析日志文件的结构, Jones 等人解读了 IE 浏览器缓存 index.dat 文件的结构以及研究从 IE 浏览器中获取已删除记录, 同时他还开发出 Pasco 工具来自动解析 index.dat 文件. Pereira1 详细介绍了火狐浏览器从第二版升级为第三版后它的历史存储系统的变化, 并且提出了一种改进的方法在磁盘未分配区域来搜索已删除记录数据, 针对的是火狐浏览器利用的 SQLite 数据库产生的临时回滚日志文件, 通过对已删除回滚日志文件的提取, 能够有效地恢复出取证研究者感兴趣的信息数据.

① 收稿时间:2013-05-27;收到修改稿时间:2013-06-20

3 技术介绍

从技术角度来说, 取证人员为了获取犯罪嫌疑人在使用电子产品时留下的某些作案信息或证据, 所要分析的存储数据介质可能有多种, 它可能是嫌疑人使用过的硬盘、光盘、软盘、压缩硬盘、U 盘、内存以及其他一些可以存储数据的介质. 计算机取证一般分为四个步骤: 保护, 证实, 恢复分析以及归档备份, 取证方法一般是通过专业取证软件和硬件来实现, 通过一些预处理, 分析检测计算机系统、数据结构进而恢复并且保护证据. 如表 1 所示, 介绍了一些实用的 WEB 取证工具.

表 1 WEB 取证工具

名称	版本	来源	描述
ChromeAnalysis	1.0.1	forensic-software	针对 chrome 浏览器历史数据分析
ChromeCacheView	1.26	Nirsoft	分析 chrome 浏览器缓存文件
FoxAnalysis	1.4.2	forensic-software	分析火狐浏览器产生的历史数据
IECacheView	1.36	Nirsoft	获取 IE 浏览器缓存文件
IEHistoryView	1.56	Nirsoft	获取 IE 历史浏览 URL
MozillaCacheView	1.36	Nirsoft	获取火狐浏览器的缓存数据
PasswordFox	1.30	Nirsoft	获取火狐浏览器中用户的姓名和密码

本文主要针对于 web 浏览器取证方法与技术作一些介绍, 其中包括 IE、MozillaFirefox 等一些主流的浏览器. web 取证的方式各有不同, 可以通过系统缓存文件进行恢复, 如 IE、chrome、firefox 的 cache 文件、index.dat 文件; 也可以通过浏览器系统本身机制产生的临时文件进行恢复, 如日志文件等. 由于不同的浏览器具有不同的数据存储方式, 因此不同的浏览器需要采取不同的数据分析方法. 当 cache 文件以及日志文件被计算机操作系统或是犯罪分子所删除的时候, 只要磁盘管理系统尚未进行重新分配磁盘空间, 写入的数据未覆盖原有数据, 磁盘上依然存在着这些已删除文件数据, 而这些数据可能存在于磁盘的未分配存储空间上, 我们就可以通过读取未分配磁盘区域物理镜像来恢复或提取数据.

通过实验的方式来找取证信息可能存在的位置, 这也是一个比较实际的方法, 实验先决条件是你必须设定一些重要关键词, 用关键词检索网页信息, 然后用硬盘写保护方式, 阻塞硬盘的写入, 最后用性能较高的字符串匹配算法来匹配所设定的关键词, 由此来发现取证信息的存在方式和具体分布的位置, 这里面需要考虑到关键词的编码方式, 编码不匹配会导致找不到具体的查询结果.

总结 Web 取证方式:

① 根据已删除浏览器 Cache 缓存文件进行恢复;

② 根据已删除浏览器日志文件进行恢复;

③ 根据已删除浏览器数据库临时文件进行恢复;

④ 根据操作系统的 page.sys 文件进行恢复.

本文将重点介绍根据数据库文件及其临时日志文件进行数据恢复的方法和技术, 以及对浏览器 Cache 缓存文件的取证和分析.

3.1 火狐浏览器取证技术

3.1.1 SQLite 数据库文件

sqlite 数据库是嵌入式设备常用的一种轻量级的数据库, 同时也被用于一些浏览器存储历史记录, 火狐浏览器使用的就是 sqlite 数据库. 由于不同的浏览器存储数据的文件结构会有不同, 因此我们得通过实验找出其特征点, 依据特征来发现文件信息. 比如说文件的魔术字段, sqlite 就有其头部魔术特征, 如图 1 所示.

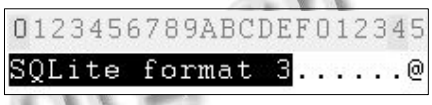

图 1 Sqlite 魔术字段

还可以通过字符串匹配算法, 来按字节流进行匹配, 如果是要恢复 URL, 它的特征字符串是“http: //”、“ftp://”等一些具有明显区分度的关键词组, 如图 2 所示.

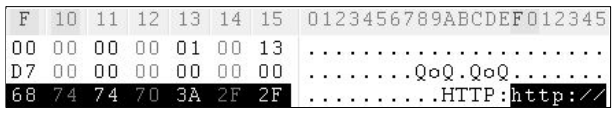

图 2 “http://”特征字段

火狐浏览器使用了多个 SQLite 数据库, 这些数据库文件存在于用户文件夹中, 具体的文件位置, xp 系统 C:\Documents and Settings\<user>\Application Data\Mozilla\Firefox\Profiles\<profile folder>\; Vista 系统或 windows7 系统 C:\Users\<user>\AppData\Roaming\Mozilla

\Firefox\Profiles\<profile folder>\每个数据库文件都是以“.sqlite”扩展名结尾，每一个数据库都有一个独立的“.sqlite”文件. 我们可以通过使用火狐数据库插件来实现对数据库数据的管理，SQLite Manager program(http://code.google.com/p/sqlite-manager)或者是 SQLite 命令行工具(http://www.sqlite.org/download.html)来对数据库进行操作，对于用户来说，可以通过输入 URL 的任何一部分，火狐浏览器将会自动补全并且提示历史存储的URL，这将大大提高效率.

3.1.2 火狐浏览器 Places.sqlite 数据库

Places.sqlite 数据库具有多张数据库表，下面主要介绍与浏览历史密切相关的两张数据库表格.

(1) Moz_places 表

Moz_places 表是 Places.sqlite 数据库的核心数据库表，它存储了用户访问网页的所有 URl 地址信息，Moz_places 表包含以下重要字段:

① id: 表编号字段，自增;

② url: 存储访问过网页的 URL;

③ title: 存储网页的标题;

④ rev_host: 反向存储主机名;

⑤ visit_count: 存储网页的总访问次数;

⑥ hidden: 指明 URL 是否被隐藏，为 1 被隐藏，为 0 不被隐藏;

⑦ typed: 指明URL是否被标记，为1被标记，为0 不被标记;

⑧ favicon_id: 与 moz_favicons 表 ID 值相联系;

⑨ frecency: 访问频率得分，得分越高表示此网页最常访问.

moz_places 表存在着某些取证兴趣点:

a) 系统默认存储 URL 时效为 90 天，可能会跟版本的不同而改变;

b) Typed 字段标记为 1 表示用户手动输入 URL;

c) Frecency 字段结合 visit_count 和 typed 字段，决定用户行为.

如图 3 所示，为 places.sqlite 数据库中 moz_places 表数据视图.

id	url	title	rev_host	visit_count	hidden	typed	favicon_id	frecency	last_visit_date	guid
1	http://www....		moc.allizom....	0	0	0		131		9u_t6OfcTOgG
2	http://www....		moc.allizom....	0	0	0	1	131		P01VewLt01Wr
3	http://www....		moc.allizom....	0	0	0	2	131		-hh6Oo5xha00
4	http://www....		moc.allizom....	0	0	0	3	131		7aFsi53zTPct
5	http://www....		moc.allizom....	0	0	0	4	131		R3flRfBiJsck
6	place:sort=8...			0	1	0		0		eK7a5tVV2WXi
7	place:folder=...			0	1	0		0		j22UlCmdsHcq
8	place:type=6...			0	1	0		0		HeG5LNsWGG7y
9	http://www....		moc.allizom....	1	1	0		94	1365994448...	KFWNYvsCvJN6
10	http://start.fi...	火狐主页起始页	nc.anihcxofer...	11	0	0	6	1100	1366262606...	guaXJP2MQduP
11	http://www....	欢迎使用 Fire...	gro.allizom.w...	1	0	0	5	94	1365994453...	D3SPOjaThREg
12	https://addo...	Firefox 附加...	gro.allizom.s...	1	0	0	7	94	1365994472...	6Slbe7X8Sc2d
13	https://addo...	sql :: 搜索 :: Fi...	gro.allizom.s...	1	0	0	7	94	1365994522...	0V0FHrrRxq5v
14	https://addo...	Firebug :: Fir...	gro.allizom.s...	1	0	0	7	94	1365994566...	YBwbVnAXEmX8

图 3　moz_places 表数据视图

(2) Moz_historyvisits 表

Moz_historyvisits 表是 Moz_places 数据库中另一个重要的表，它存储了历史的浏览数据，访问类型，以及 URL 指针，即访问网页的一个顺序，其中包括了一下的一些字段:

① id: 表编号字段，自增;

② place_id: moz_places 表中的 id;

③ form_visit: 上一条访问网页 URL 指针;

④ visit_date: 访问网页时间;

⑤ visit_type: 访问状态标记;

⑥ session: 存储网页的 session ID.

通过火狐 SQLite 扩展管理器进行如下查询:

“SELECT historyvisits.id, historyvisits.from_visit, historyvisits.visit_type, historyvisits.place_id, places.url from historyvisits, places where historyvisits.place_id = places.id”如图 4 所示.

以第二条记录为例它的 from_visit 字段的值为 1，这说明它的上一跳网页是 id 值为 1 的 URL，即: http://start.firefoxchina.cn/，通过这个我们很清楚地知道用户在用浏览器上网的过程中的访问网址的情况，这就是取证过程中的用户行为分析. 但是，用户很可能会通过浏览器自带的历史记录清楚功能把历史记录给删除，或是直接将数据库文件删除，这就给取证人员带来了烦恼，因为通过浏览器删除历史数据的过程

id	from_visit	visit_type	place_id	url
1	0	1	9	http://start.firefoxchina.cn/
2	1	1	10	http://i.g-fox.cn/search?q=&en...
3	2	6	11	http://i.g-fox.cn/rd2.html#q=&en...
4	3	1	12	http://www.baidu.com/baidu?tn=...
5	0	1	13	http://www.baidu.com/index.php?...
6	5	1	14	http://tieba.baidu.com/
7	6	6	15	http://tieba.baidu.com/index.html
8	7	1	16	http://tieba.baidu.com/p/2173144...
9	0	1	9	http://start.firefoxchina.cn/
10	0	1	9	http://start.firefoxchina.cn/
11	0	1	17	http://i.firefoxchina.cn/search?q=...
12	11	6	18	http://i.firefoxchina.cn/redirect/sea...
13	12	1	19	http://s8.taobao.com/search?com...
14	13	6	20	http://www.taobao.com/go/chn/t...
15	14	1	21	http://redirect.simba.taobao.com/r...
16	15	6	22	http://www.taobao.com/go/chn/t...
17	0	1	9	http://start.firefoxchina.cn/
18	17	1	10	http://i.g-fox.cn/search?q=&engin...

图 4 联合查询结果视图

是不可恢复的过程, 它对文件数据进行了清零处理或是重写, 对文件的原数据造成了破坏. 文件恢复的宗旨是文件的原数据未遭到破坏, 这就需要取证人员采取新的恢复技术来达到目的. 由于不同的浏览器的存储历史数据的机制不同, 这就需要取证人员具体情况具体分析, 这里就以火狐浏览器为例介绍一下数据的获取. 前面已经介绍火狐浏览器的历史数据存储系统, 它是通过 SQLite 数据库来实现的, 即使用户通过浏览器的清理历史记录功能来删除数据库中的数据, 我们依然可以通过数据库的临时文件来恢复数据. 火狐浏览器在使用 SQLite 数据的时候会出现一个回滚日志文件, 它是一个以“-journal”结尾的文件, 在每次事务开始的时候产生, 在记录提交后删除, 这就给了取证人员启示, 在磁盘的未分配区域可能存在着一些临时回滚日志文件的原数据. 可以恢复这部分数据来达到取证的目的, 并且火狐浏览器的临时回滚日志文件的数据存储方式同数据库文件的方式. 因此在恢复的时候应该分析文件的存储方式, 这个过程将有助于高效地获取有效的信息.

3.1.3 SQLite 数据库存储结构以及恢复技术

SQLite 数据库的数据结构为带索引的 B+树(如图 5 所示)或者 B-树(如图 6 所示).

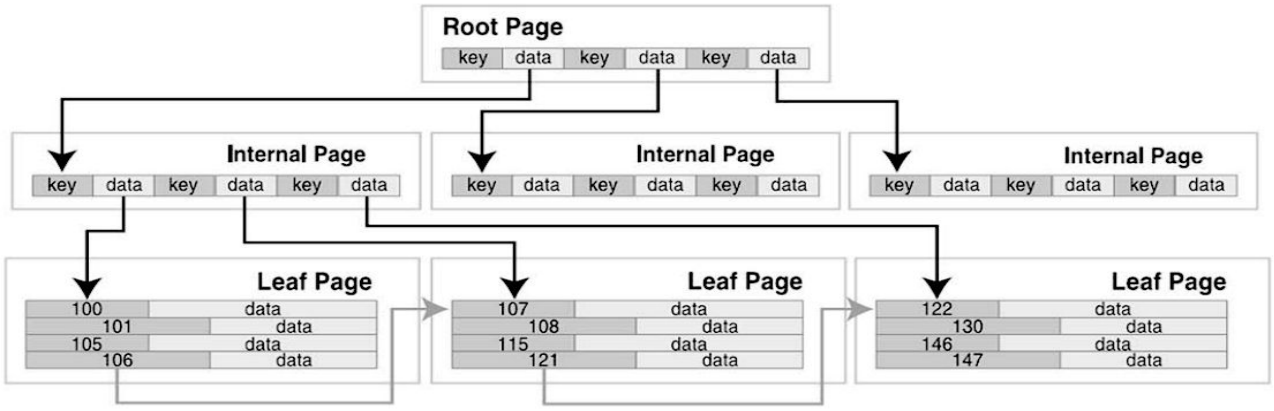

图 5 B+树结构图

树的内部节点不存储具体的数据, 只作为子节点的索引, 具体的数据将会被存在树的叶子节点上, 数据存储的逻辑结构如图 7 所示.

头部包括 hsize 和数据大小, 以及 T1-TN 一组表示数据类型和大小的数值, 这组数值描述了 Data segment 数据部分的数据类型和大小. 通过解析数据结构和解码分析, 使得数据能够高效地提取. 如图 8 所示, 很清晰地分析出数据记录的存储格式, 这将大大提升取证效率.

B-树结构

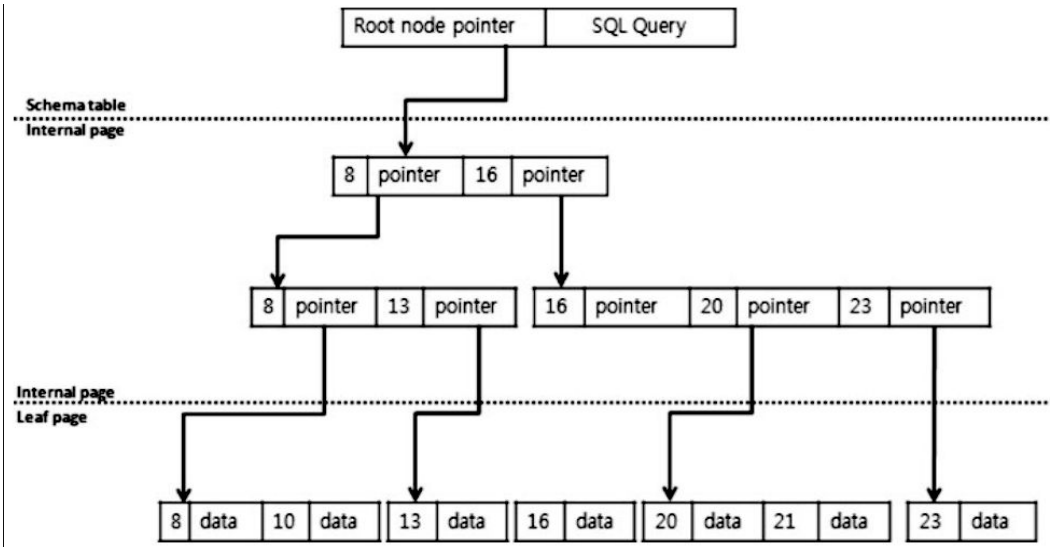

图 6　B-树结构图

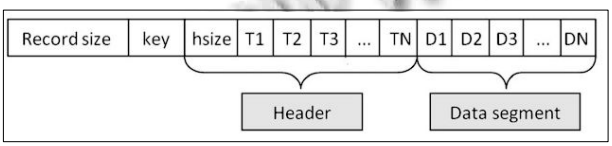

图 7　逻辑结构图

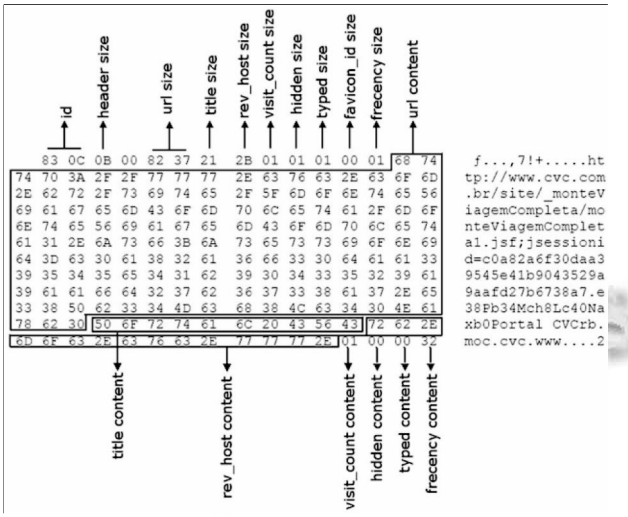

图 8　记录数据结构解析图

在记录恢复的过程中，可以通过字符串匹配算法来实现具体的有效记录的提取，如通过检索匹配“http”字符串，即 OX 68 74 74 70 十六进制数值，再通过提取 size 值来确定记录的大小，进而来恢复 URL 记录.

以上介绍的就是通过火狐浏览器产生的临时回滚日志文件来恢复数据信息的方法，这种方法可以举一反三，虽然不同的浏览器会有不同的存储数据机制，但是有可能像火狐浏览器这样产生临时数据库日志文件等其他的一些临时文件，如 IE 浏览器会产生 daily/monthly index.dat 文件，火狐的 cache 文件，chrome 同样会产生数据库临时文件. 差异就在于他们的数据结构可能会不同，这就需要我们进一步去研究临时文件的数据存储结构，根据不同的结构特征，再根据字符串匹配算法来恢复想要的那部分数据.

下面将介绍根据 IE 浏览器缓存文件来恢复数据的方法，缓存文件中保存着很多有效的信息，包括网页元素，如图片，网页文本，网页 html 文本等等.

3.2 IE 浏览器取证技术

IE 浏览器历史记录数据存储在 index.dat 文件中，其中包括 Cookie 数据，历史记录和缓存文件，index.dat 文件会存储在不同的子目录中. 当 index.dat 文件被删除时，原数据不会清除，而是会被移动到磁盘未分配区域，只要数据没有被重写，就可以恢复出完整的历史数据记录.

3.2.1 存储结构介绍

IE 浏览器历史数据记录有 3 种类型，分别是 URL，LEAK，REDR，每条记录的开头四个字节指明了记录的类型. Cookie 记录和历史记录为 URL 类型，而 Cache 记录可以具有 3 种类型中的任何一种. URL 和 LEAK 类型包含了丰富的信息，如记录类型、记录长度、最后修改时间、用户名、网页信息以及缓存文件路径信息，而 REDR 类型只包含了单一的网页数据，数据结构如表 2、表 3 所示.

URL 或是 LEAK 类型数据结构:

表 2 URL、LEAK 类型结构

字节偏移量	字节	描述
0x00	4	URL 或 LEAK 魔术字节
0x04	4	长度字段
0x08	8	最后访问时间
0x10	8	最后修改时间
0x34	4	数据记录开始字节
0x3C	4	缓存文件标记

REDR 类型数据结构:

表 3 REDR 类型结构

字节偏移量	字节	描述
0x00	4	REDR 魔术字节
0x10	*	网页数据

3.2.2 Cookies

每一条 cookie 记录以“URL”关键词开始以及一个标记关键词“Cookie”, 以此来区分历史数据和缓存数据. 如图 9 所示.

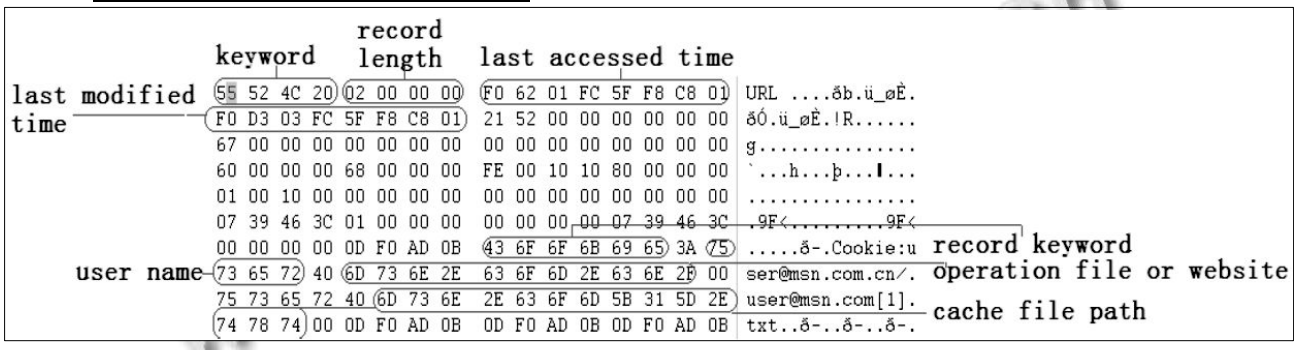

图 9 cookie 记录格式解析

Keyword 表示类型, 0x55 52 4C 20 即 URL 类型; record keyword 表示记录关键字, 0x43 6F 6F 6B 69 65 即 Cookie. 这些数据表明这是一条 cookie 记录, 一条完整的记录包含了记录的长度、访问时间、用户名以及缓存文件路径等等一些重要的信息.

3.2.3 历史记录

历史记录不仅记录了用户访问网页的信息, 还记录了用户打开本地磁盘文件的行为操作, 历史记录以“URL”关键词开头, 包含一个标识历史记录的关键词“Visited”, 如图 10 所示.

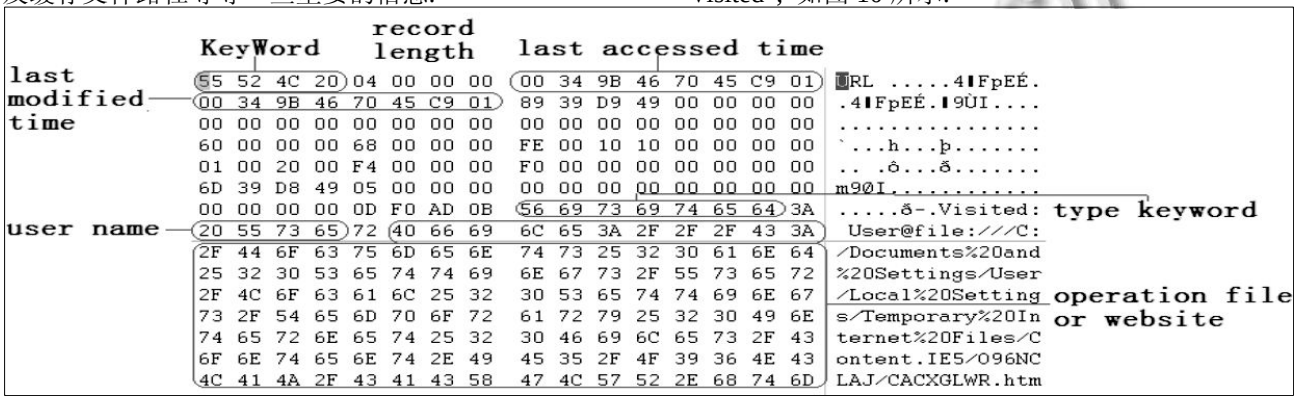

图 10 历史记录结构解析

Keyword 表示类型, 0x55 52 4C 20 即 URL 类型; type keyword 表示记录关键词标记, 0x56 69 73 69 74 65 64 即 Visited. 这些数据唯一标识一条历史数据记录, 其中包括其他的一些信息, 包括网页的元素、用户名、访问时间、记录长度等. 这些将有助于取证人员分析用户的行为.

3.2.4 缓存记录

缓存记录的结构是最为完整的, 记录了最多的信息. 在磁盘的未分配区域, 记录中包含了唯一标识记录的关键词“http”, 它指明了记录是缓存记录. 缓存记录有 3 种类型, 即 URL、LEAK、REDR, 当信息中包含 READ 类型关键词时表示只有网页地址信息, 而信

息中包含 URL 或是 LEAK 类型关键字时，记录信息就比较多，包括用户名、网址、最后访问时间、最后修改时间、点击数、部分缓存文件路径、网络协议、缓存文件类型以及缓存文件长度等等. 具体文件分析结构，如图 11 所示.

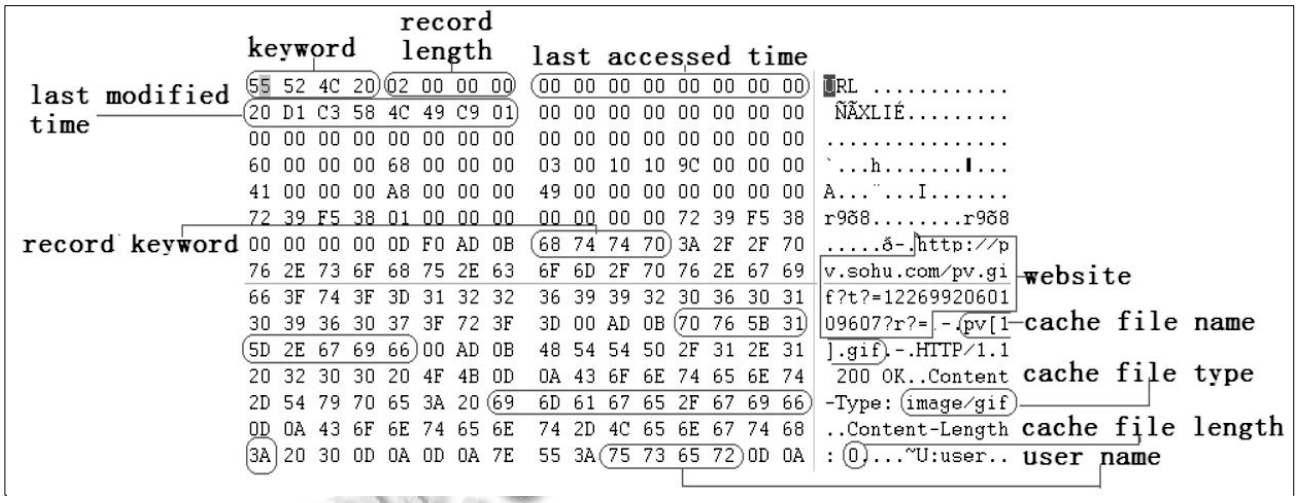

图 11　cache 记录结构解析

缓存记录的数据记录了完整的网页 URL 信息，以“http://”字符串开头，从上图可以看出用户访问“http://pv.sohu.com/pv.gif?t?=1226992060109607?r?” 的网址信息，在 cache 文件类型字段说明了这是一个图片.

3.2.5 取证技术

取证技术的首要任务是如何获取数据，网页记录数据是一些二进制数据流，我们可以直接以字节流的方式来读取磁盘的未分配区域，考虑到有 3 种记录类型，必须分为 3 个部分来分析数据，分别是 cookie 记录分析，历史记录分析，缓存记录分析. 通过记录的关键词域(“Cookie”、“Visited”、“http”)，我们可以区分不同类型的记录，进而解析数据. 通过关键词匹配算法来得到取证人员感兴趣的数据，如 URL，搜索引擎检索的关键词等.

3.3 新标签页与网页缓存取证

3.3.1 Firefox moz-page-thumbs 取证

目前主流浏览器都提供了一种新的机制，火狐浏览器称它为 Firefox moz-page-thumbs，在我们浏览网页的时候，浏览器会去当前页面的一个快照，以图片的方式保存，同时存储了这个网页的标题和 URL 信息，当用户打开一个新标签页的时候，会看到一些以图片方式展示出来的图片 URL，通过点击快照图片，就能链接到相应的网页上，用户经常访问的网页一定会显示在新标签页上，如图 12 所示.

图 12　火狐新标签页视图

这些快照中包含了快照的 URL、主题信息，这对取证研究人员来说无疑又是一个研究方向，因为这些信息都是用户在操作浏览器时候留下的，而且是用户最常访问的一些网页信息，这对分析用户行为有很大的帮助.

3.3.2 网页缓存取证

不同的浏览器的缓存机制是有所区别的，研究人员需要针对特定的浏览器机制展开研究. 浏览器的缓存文件包含了丰富的信息，存留的时间十分的短暂，即便这样我们也可以获得有效的信息，比如在缓存文件删除后，在磁盘的未分配区域，就可能存在这些缓存文件的原数据，缓存文件中保存了访问网页的完整信息，包括图片、网页数据、网页 js 代码、css 代码都会存在，如图 13 所示.

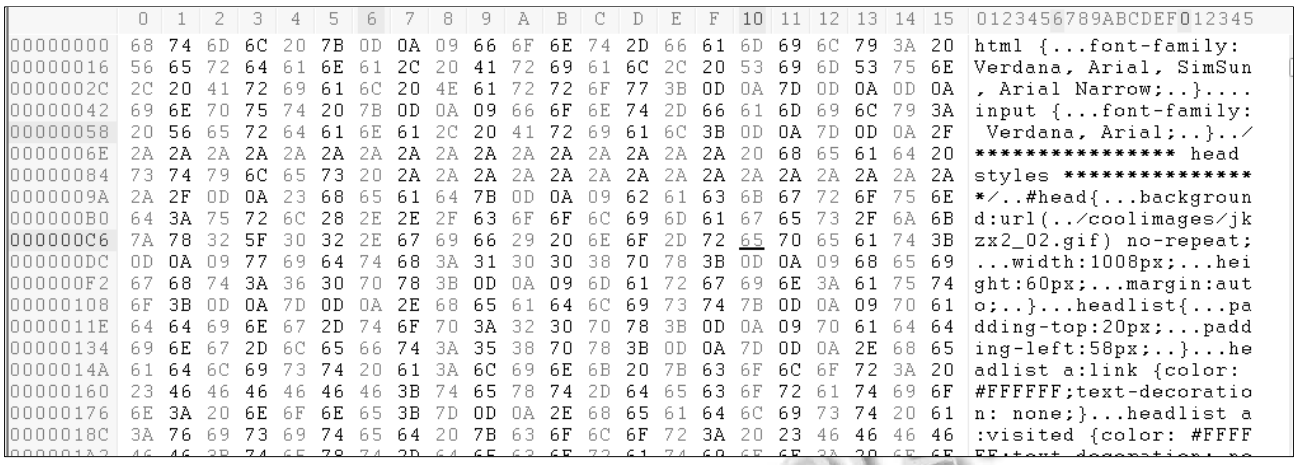

图 13 火狐缓存网页缓存文件数据视图

这是火狐浏览器的缓存文件，它包含了完整的网页文本信息，所有的网页元素都会有记录，取证人员可以通过提取这些信息来恢复用户访问过的网页 html 文本，网页上完整的数据都会存在于 html 文本上.

4 总结与展望

通过本篇文章的介绍，我们对浏览器的取证技术有所认识，由于浏览器本身的一些机制，目前浏览器的取证方法大多停留在对浏览器临时文件的恢复和取证，通过解析临时文件的结构来知道数据存储结构，通过匹配关键词字符串来获取感兴趣的信息. 针对浏览的取证工作，取证人员正在绞尽脑汁地探索更高效，信息获取更全的方法来获取有效的信息. 随着众多主流浏览器版本的不断更新，新的浏览器机制也正在不断提出，我们应该与时俱进了解新的浏览器机制，里面可能蕴含着丰富的数据记录，不要仅仅局限在有限的研究范围内，要针对新的机制提出一些有效的方法.

浏览器取证技术还有很多可以发掘的点，只要研究者善于发现隐藏的信息，随时关注主流浏览器的发展情况，必定会有所收获，而且我们不应该仅仅局限在 PC 机上，现在智能手机，移动终端的兴起，这些更值得我们去探索研究.

参考文献

1 Pereira MT. Forensic analysis of the Firefox 3 internet history and recovery of deleted SQLite records. Digital Investigation, 2009, (5): 93–103.

2 Chen L. Computer Forensics. Wuchang: Wuhan University, 2007: 1–13.

3 FirefoxForensic.Firefoxmoz-page-thumbs. http://kb.digital-detective.co.uk/.

4 Oh J, Lee S, Lee S. Advanced evidence collection and analysis of web browser activity. Digital Investigation, 2011, (8): 62–70.

5 Jones KJ. Forensic Analysis of Internet Explorer Activity Files, 2003.