IE Internet Information Forensics Technology in Unallocated Disk Space

Chen Haiping, Luo Delin*, Gao Qinquan, Qian Zhicong, Wu Shunxiang
Department of Automation
Xiamen University
Xiamen, China
Chenhp177@126.com

Abstract—**This paper presents the main content of IE (Internet Explore) Internet record data, what fields are of forensic interest and what information the available tools can extract. It shows that remnants may still be found in unallocated disk space even the IE Internet record normal file data is deleted. This paper proposes a string pattern matching algorithm to find IE Internet record data in unallocated space based on known internal record structures. The system developed can recover deleted IE records and it may be an approach to obtain evidence in unallocated disk space.**

Keywords-forensics; IE Internet records; string pattern matching

I. INTRODUCTION

The rapid development of the Internet technology today has brought great convenience to our daily life, but also provides new criminal means. Computer information systems have become tools or criminal targets in more and more criminal activities. It's necessary to use computer forensics technology in detection of these cases, to search for evidence in order to confirm the offender and the crime, and then to engage in legal proceedings [1-2].

From a technical point, Computer Forensics is to analyze hard disks, CD-ROMs, floppy disks, Zip disks, U disks, memory buffer and any other forms of storage media in the acquisition of criminal evidence. That is, computer forensics includes the protection, confirmation, retrieves and archiving of the computer evidence stored in magnetic medium. The Forensics method usually includes the use of software and hardware, according to some pre-defined procedures, comprehensive examination of computer system to retrieve and protect evidence [3-4]. This article research into the Web browser record data, and analyze its retrieving and parsing process.

The normal data of IE browser record data include Cookie records, History records and Cache records which are saved as files are all called the same name "index.dat" in different directories. When index.dat is deleted, the information isn't just cleaned up but moved into the unallocated space in users' computers [5]. As long as these data has not been completely covered, it is possible to retrieve a complete record of Internet information.

This Project is Supported By the Planning Project of the National Eleventh-Five Science and Technology(2007BAK34B04) and the Chinese National Natural Science Fund(60704042) and Aeronautical Science Foundation (20080768004) and the Program of 211 Innovation Engieering on Information in Xiamen University (2009-2011)

II. IE INTERNET RECORDS DATA STRUCTURE

A. The Common Structure

IE Internet records data has URL, LEAK, REDR three types, which are represented in the beginning four bytes of each record [6]. The Cookie records and History records are only saved in URL type, while there are Cache records of all three types.

In both URL and LEAK types, each record contains an abundance of information, such as record type, record length, last modify time, last access time, the user name, Internet web sites and cache file path information [7]. But, in REDR type, there is only simple Internet Web sites data. The data structure is showed in the following tables.

TABLE I. URL OR LEAK RECORD DATA STRUCTURE

Offset	Bytes	Representation
0x00	4	URL or LEAK Keyword
0x04	4	The Length field，give the number of 0x80 byte blocks that make up the Internet Web site data
0x08	8	last accessed time
0x10	8	last modified time
0x34	4	The beginning of the main record data
0x3C	4	A sign to cache file name（only cache records have）

TABLE II. REDR RECORD DATA STRUCTURE

Offset	Bytes	Representation
0x00	4	REDR Keyword
0x10	--	Internet Web site data

The following URL type record is taken for an example to show the differences among Cookies, History records and Cache records.

B. Cookies

Every Cookie record begins with the keyword "URL" and a sign record keyword "Cookie" which distinguished from History records and Cache records.

978-1-4244-5273-6/09/$26.00 ©2009 IEEE

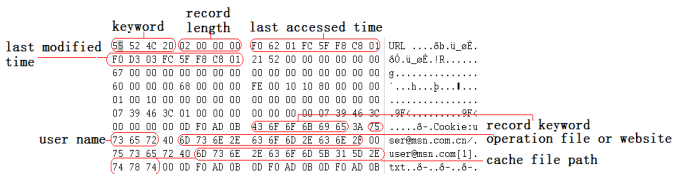

Figure 1. A Cookie record in the unallocated space

Corresponding with the data structure table I, the table below is formed.

TABLE III. THE EXPLANATION OF THE DATA IN FIGURE 1

Offset	Hexadecimal Data	Explanation	Field Data
0x00	55 52 4C 20	The keyword	URL
0x04	02 00 00 00	Record length	2 multiplied by 0x80 , it's 256 bytes
0x08	80 FC FB 11 20 3F C9 01	last accessed	
0x10	20 8F 89 66 6E 45 C9 01	last modified	
0x34	68 00 00 00	Website begin	0x68

The main website data begins at offset 0x68 from the beginning of the record. The description, "Cookie: user@www. microsoftaffiliates.net/", shows that it's a cookie record, the website is "www.microsiftaffiliates.net" and the user name is "user" bounded by the character "@". After that, it comes up with a string "user@msn[2].txt" which is a file name of Cookie's cache file. This file contains cookie's specific key value of the website "www.microsoftaffiliates.net".

C. *History Record*

History records not only record the user access information, but also recorded the operating procedures of users' opening some other documents in the local hard disk. History records also begin with the keyword "URL" and a sign record keyword "Visited". There's an example followed.

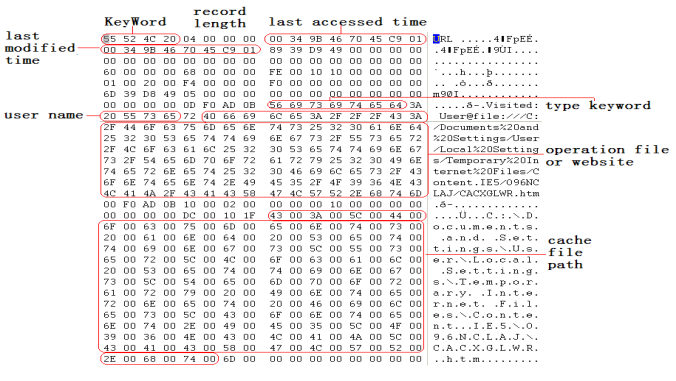

Figure 2. A complete history record in the unallocated space

As in figure 2, History records data starts with the record keyword "Visited". The user's name "User" also is bounded by the character "@". There is a cache page that the user had accessed. History records' cache file address is not recorded at offset 0x3C as in Table 1 as Cache record, but in the data at the end of 0x14 bytes from record beginning. The cache file address data is encoded in Unicode, which are two-byte characters.

D. *Cache Record*

Cache records structure is the most complex, recorded most information and used frequently among the three records' types. In unallocated space, the record data with a sign record keyword as "http" is thought to be Cache record data.

Cache records have URL, LEAK, REDR three types. While the information contained in REDR type's data is only of website address, URL and LEAK records record the respective user name, the network address, the last access time, last modified time, hits, part of cache file path data, as well as the HTTP protocol, web cache file types (images, text, etc.), web cache file length field, etc.

At offset 0x20 in normal file index.dat, there are strings of four cache record folders path. However, in unallocated disk space, it probably cannot be recovered, and the cache file cannot be located.

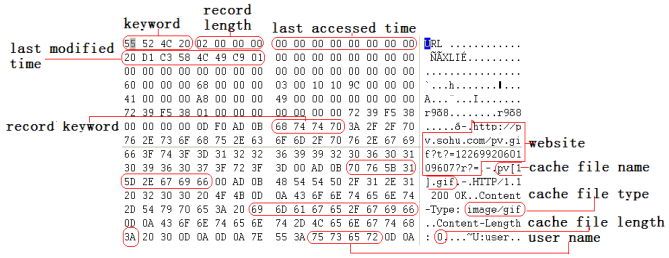

Figure 3. A cache record in the unallocated space

Cache record data records a complete network address which the user used to open. With the beginning string "http://", the figure shows the network address "http://pv.sohu.com/pv. gif?t?=1226992060109607?r?=". The string "pv[1].gif" is a cache file name of the special website and ".gif" means it's a picture. After then, there is the cache file type, the cache file length, the user name and maybe other information.

As deleted data may not be recovered completely, the searching HASH table method of analyzing index.dat normal file data is no longer fit for unallocated disk space. Only through searching keywords can we locate the exactly record data. In this paper, the pattern matching algorithm used for searching is an algorithm called KMP.

E. *The Pattern Matching Algorithm KMP*

String search is a basic operation. C library functions have already provided the "strstr ()" function which is a linear search for string search. When the character does not match the pattern, the point will move on just a byte after the current position and continue to search.

KMP matching algorithm will find the most suitable location to retire when it fails, rather than simply return back to the first character of the substring. In order to find the appropriate location, the first string must be pre-treated to build a back array. The operation according to the back array carried out to be much more efficient while compared to linear search [8].

III. IE INTERNET INFORMATION RETRIEVAL SYSTEM DEVELOPMENT RECORDS

A. System Analysis

In system design, the first thing to do is to fetch data. Websites Record data are some binary data stream. In program design, we can directly read every sector in the unallocated disk space. The searched data stream using BYTE data type should be stored in the memory buffer.

Considering there are three types record data, the system could be divided into three sub-modules: Cookie Records Analysis, History Records Analysis and Cache Records Analysis. Through the sign record keyword ("Cookie", "Visited" and "http"), different type records can be distinguished from others by switch cases in programming. As different records analyses are of the same principle, the programming process should be alike. Here records with the keyword "URL" are taken for instance.

In main programming, search for the keyword "URL" first. If found, then search for the record keyword "Cookie". If found, it's a Cookie record and run into the Cookie Records Analysis module. Else if "Visited" is found, it's a History record and chooses the History Records Analysis module for analyzing, and so on.

In each module, the searched record data should be parsed and the properties in the figure 1, 2 or 3 should be gathered and saved in a record data structure. And at last, these data will be displaced in some understandable form.

As the unallocated space of user's computer is usually very large (up to 10G and above), searching data of such a big space must need a long time. Therefore, in design, the efficiency of system must be paid much more attention to. So in programming the parameter type "CString" should be replaced by "wchar_t *" pointers which is more efficient.

B. System Design

The exact system design is explained in the system analysis above. And the main idea of the analyzing system is showed in the figure below [10].

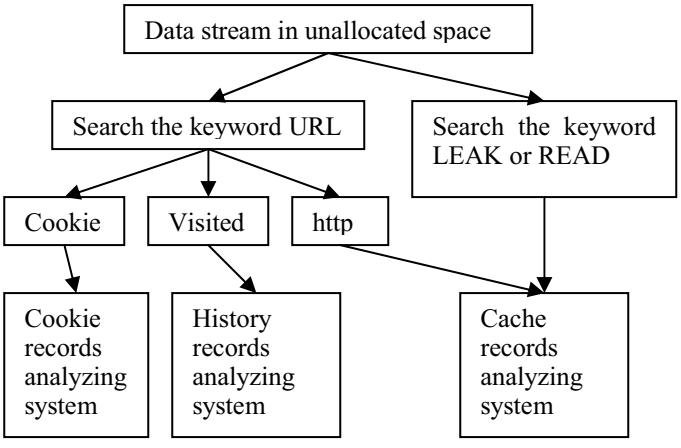

Figure 4. The main idea of the analyzing system

C. System Implementation

The system runs on Windows XP operating system, the main choice of development tools is Visual Studio 2005.

The definition for the data structure in C++ format:

```
struct ResultRecode
{
    CString strUserName;          // user name
    CString strVisitTime;         // last accessed time
    CString strModifyTime;        // last modified time
    CString strURL;               // network address
    CString strCacheFile;         // the cache file path
    ResultRecode *pNextRecord;    // pointer to next node
    ResultRecode *pPreRecord;     // point to previous node
    ResultRecode *pParent;        // pointer to parent node
};
```

Three sub-modules package interfaces are defined like this:

```
ResultRecode *AnalyzeCookie (BYTE *bData,
                             DWORD dwDataLen);
ResultRecode *AnalyzeHistory (BYTE *bData,
                             DWORD dwDataLen);
ResultRecode *AnalyzeCache (BYTE *bData,
                             DWORD dwDataLen);
```

The internal processes of the three modules are basically the same. Take the cache record with URL keyword for an example to show the internal code procedure in Figure 5.

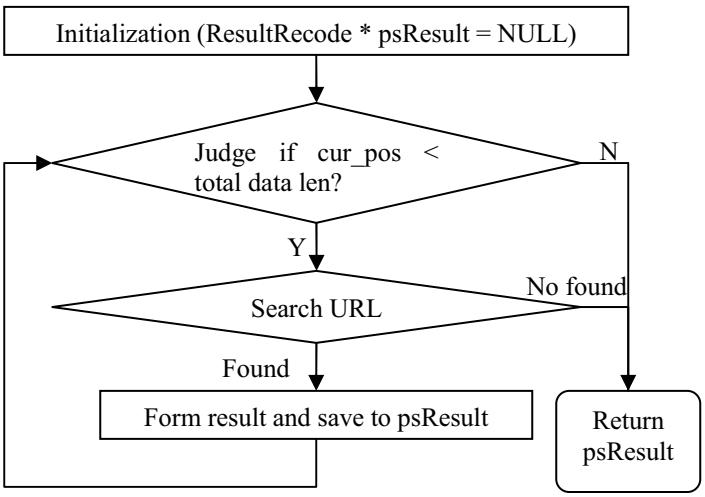

Figure 5. The internal process

Sub-module parameters of the entrance are defined to be BYTE type data stream. Thus, the records data analysis, whether in normal file or unallocated space, can use the same function interface. Without redefining interface, the system compatibility is improved.

D. KMP in Programming

The basic operation of this system is the use of KMP algorithm technology to locate the Internet records data. In programming, there must first adds "KMP.h", "KMP.cpp" document and the code "#include KMP.h", so that KMP algorithm can be used conveniently in code operation in the form of "CKMP" class. That can greatly reduce the complexity of the code. KMP algorithm in the concrete application is like this:

```
CKMP kmp;                    //define a class
int pos = 0;
 // keyword URL
BYTE pUrlKeyword [] = {0x55, 0x52, 0x4C, 0x20};
// to store the searched data offset
int KeyWordOffset = -1;
while (pos < dwDataLen)
{
      int iFind = kmp.ResKMP ((char *) (pData + pos),
                        dwDataLen, (char *)pUrlKeyword,
                        4, KeyWordOffset);
      if (iFind){
      // find the record data.
      ……
      }
}
```

The parameter "iFind" means whether a record is found. If iFind != 0, the record offset is saved in the parameter "KeyWordOffset" which means the distance from the pointer "pData".

E. Figures

The system designed above runs to search the allocated disk space of logical partition C as default, and uses a simple MFC open file dialog to choose in which logical partition the unallocated space is. A Cache record example of system operation is showed as follow:

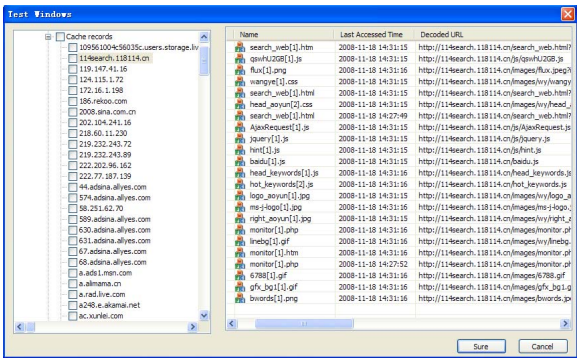

Figure 6. Cache records

IV. CONCLUSION

In this paper, we do a detailed analysis and comparison on the data stream of IE Internet Cookie records, history records and cache records. And develop an analyzing system of the websites records information in unallocated space based on a string pattern matching algorithm. This system can be used as a part of computer forensics software.

ACKNOWLEDGMENT

The authors of this paper would like to thank the anonymous reviewers of this paper for their carefully reading of the manuscript as well as their many helpfully suggestions and corrections.

REFERENCES

[1] Steele. Windows Guide to Computer Forensics Investigation [M]. Wu Yu, Tang Hong, Chen Long. Beijing: Science publisher, 2007: 239-242.

[2] Chen Long. Computer Forensics [M]. Wuchang. Wuhan University, 2007: 1-13.

[3] Xu Rongsheng. Development of Chinese digital Forensics. [J]. China Education Network, 2007, (8).

[4] Ding Liping, Wang Yongji. Computer forensics techniques and research tools [J]. Information security and confidential communications, 2005, 8.

[5] Huang Xuan. Ma Yanli. Zhao Zhansheng. Data Forms in Information Security [N]. NET Security Technologies And Application. 2002, 8: 12-16

[6] Keith J. Jones. Forensic Analysis of Internet Explorer Activity Files [EB/OL]. 2003.

[7] Keith J. Jones. Forensic Analysis of Microsoft Internet Explorer Cookie Files [EB/OL]. 2003.

[8] Bao Zhengrong, Wang Yongcheng, Liu Shengong,etc. A fast cross string pattern matching algorithm [J]. Journal of Shanghai Jiaotong University, 2003, 37(3): 420.

[9] Anonymous. The Study and Design of IE Trace Programs [N]. NET Security Technologies And Application. 2003, 11: 34-37

[10] Wang Ping, Feng Jianhua.C++ Object-oriented programming [M]. Beijing: Qinghua University, 2006