Deleted Data Recovery

Introduction

A critical element of web browser forensic analysis is the recovery of deleted data. HstEx is an advanced, professional forensic data recovery solution, designed to recover browser artefacts and Internet history from a number of different source evidence types.

HstEx Processing

HstEx has been designed to process a forensic image, physical/logical disk or binary dump at sector level. It does not work at the file system level. The recovered data fragments are written out to an HSTX file which can then be imported into NetAnalysis.

When HstEx searches your source, it will search it a sector (or number of sectors depending on the block size set) at a time. HstEx uses linear processing and will examine each block of data contiguously. This means that it will potentially recover data from the areas outlined in Table 1.

Potential Evidence Source			
Unallocated Clusters	Allocated Clusters	Cluster Slack	Volume Slack
Memory Dumps	Binary Dumps	Swap Files	Hibernation Files
Unused Disk Space	Live Files	Resident Files	Deleted Files
Restore Points	Shadow Volumes	Hidden Partitions	Deleted Partitions

Table 1

HstEx works in a similar way to an imager in that it starts at sector zero and processes all the data to the end. In many cases, it can recover individual records relating to browser activity without the entire file being present on the source image or disk.

As HstEx ignores the file system, it can be run across many source file system types without issue. It also means that when it recovers from a disk or image, it will potentially recover the live data as well as any that is deleted.

To identify the location of source evidence, HstEx embeds the exact location of each data fragment inside the HSTX file. NetAnalysis can interpret the exact location and present that to the forensic examiner. This allows an independent third party to verify the exact source of the evidence on the original source disk or image.

In addition to physical devices and volumes, HstEx supports all of the major forensic image formats (as shown in Table 2).

	Forensic Image Source Type	**Extension**
EnCase® v1-7 Image File (EVF / Expert Witness Format)	*.e01	
AccessData® FTK Image Files	*.e01, *.001, *.s01	
SMART/Expert Witness Image File	*.s01	
X-Ways Forensics Image File	*.e01	
Tableau Imager	*.e01, *.dd	
VMWare Virtual Disk File	*.vmdk	
Virtual Hard Disk File	*.vhd	
Segmented Image Unix / Linux DD / Raw Image Files	*.000, *.001	
Single Image Unix / Linux DD/Raw Image Files	*.dd; *.img; *.ima; *.raw	
Memory Dumps	*.dmp; *.dump; *.crash; *.mem; *.vmem; *.mdmp	
Binary Dumps	*.bin; *.dat; *.unallocated; *.rec; *.data; *.binary	
Micro Systemation Extraction File	*.xry	

Table 2

Limitations of Linear Processing

What HstEx cannot do is recover data that traverses a cluster boundary on non-contiguous clusters. This is one of the reasons why you

need to also extract and examine the available live data.

Record Based Extraction (RBE)

With many of the browser types, HstEx uses a powerful search engine which is capable of Record Based Extraction.

In some circumstances, there can be limitations with RBE. Some live browser files contain information that cannot be recovered using RBE. For example, Microsoft Internet Explorer cache records contain an integer representing a zero based index which identifies the location of the cached item. Whilst the index is contained within the record, the folder array containing the folder name string is stored at the start of the file. RBE will not recover the name of the folder as it is not stored within the record.

File Based Extraction (FBE)

Another extraction methodology employed by HstEx is File Based Extraction. Some browser index files are designed in such a way as to make RBE impossible. The History file from Firefox v1 - 2 is one such example. Firefox v1 - 2 uses a Mork database which, because of its complicated structure, makes RBE impossible. As such, it is impossible to recover individual Mork entries from unallocated clusters. In this case, HstEx employs FBE to recovery Firefox v1 - 2 History.

Recommended Forensic Methodology

We recommend that you extract the live data from your source as well as processing the entire image so that you recover potentially fragmented live files and all the recoverable deleted data.

This is because:

- HstEx employs a mixture of Record (RBE) and File Based Extraction (FBE)
- Fragmented data cannot be recovered with Linear Processing
- HstEx does not support all the data types supported by NetAnalysis
- NTFS compressed data is not processed at sector level

Of course, you will end up with some duplication during your examination, but this is a small price to pay to ensure that you have all the possible evidence.

You will also need to recover live cache files for processing so that NetAnalysis can rebuild the visited pages. Internet Explorer cache entries have an index value which points to a zero based string array which stores the cache folder name. This is stored at the front of the file. This means you have to import a live cache INDEX.DAT file to get the full original path of the cache object.

During RBE extraction used by HstEx, although we can identify the index value for the array, we do not have the string array containing the folder names; therefore it is not possible to identify full cached paths using Record Based Extraction. This is why you must use both methods for a full examination.