hp Web Browser Session Restore Forensics

– A valuable record of a user's internet activity for computer forensic examinations

Each session of activity in a Mozilla browser is recorded by the browser so that in the event of the browser crashing the session can be restored. Session Restore saves all open windows and tabs, width, height, and position of each window, scroll position within each scrollable area in each window, history of closed tabs and windows, cookies, text typed in forms and information to restart downloads. It is possible to recover multiple versions of this information in order to show a user's internet activity. Other browsers also have the facility to restore a previous session.

Introduction

In a recent examination I came across lots of urls in the unallocated space of a hard drive that were of interest and which I discovered were part of information recorded by the Mozilla browser to enable it to restore a user's session in the event of a crash. A subsequent search revealed 66 instances of full Session Restore files in unallocated space each of which could be used to show a snapshot of the browser windows and tabs that the user had open at one point in time; in addition there were many other fragments of session restore files.

Mozilla Session Restore

The MozillaWiki (1) describes the goals and objectives of Session Restore as

"After a forced restart, restore the user's workspace exactly as it was."

It goes on to state that the data saved will be –

- All open windows and tabs
- Width, height, and position of each window
- Scroll position within each scrollable area in each window
- Tab histories
- Cookies
- Text typed in forms
- Restart downloads

The session restore data is saved during the session in a file ***sessionstore.js*** and is stored on disk as a serialised JavaScript data structure, sometimes alternatively described as JSON (2) structure. At the time of writing it is stored as plain text but there is reference on the MozillaWiki to the question as to whether or not it should be encrypted for privacy reasons. The MozillaWiki also mentioned that there is a backup of the file in sessionstore.bak, but I haven't found a copy of such a file yet.

The ***sessionstore.js*** file is stored in the following location (in Vista)

C:\Users\UserName\AppData\Roaming\Mozilla\Firefox\Profiles\########.default

and is deleted when the browser closes normally.

I haven't tested this but I believe that there is a similar session restore facility in other Mozilla based browsers like Netscape, SeaMonkey, K-Meleon, Flock and others.

The ***sessionstore.js*** file, in the simplest form, looks like this –

({"windows":[{"tabs":[{"entries":[],"attributes":{}}],"selected":1,"_closedTabs":[],"_hosts":{},"width":"1628","height":"996","screenX":"33","screenY":"24","sizemode":"maximized"}],"selectedWindow":1,"_closedWindows":[],"session":{"state":"running","lastUpdate":1262210477090}})

This is with one window open with no open tabs. The notations (2) used in the file structure are as follows –

An object is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).

An array is an ordered collection of values. An array begins with [(left bracket) and ends with] (right bracket). Values are separated by , (comma).

A value can be a string in double quotes, or a number, or true or false or null, or an object or an array. These structures can be nested.

A string is a collection of zero or more Unicode characters, wrapped in double quotes, using backslash escapes. A character is represented as a single character string. A string is very much like a C or Java string.

A number is very much like a C or Java number, except that the octal and hexadecimal formats are not used.

In the example above both the name and value are enclosed in quotes but in a case I have examined from 2008 just the values are in quotes.

The following is a ***sessionstore.js*** file where just one window with one tab is open.

({"windows":[{"tabs":[{"entries":[{"url":"http://www.json.org/","title":"JSON","ID":0,"scroll":"0,0"}],"index":1,"attributes":{"image":"http://www.json.org/favicon.gif"},"_formDataSaved":true}],"selected":1,"_closedTabs":[],"_hosts":{"json.org":true,"org":true,"www.json.org":true},"width":"1628","height":"996","screenX":"33","screenY":"24","sizemode":"maximized"}],"selectedWindow":1,"_closedWindows":[],"session":{"state":"running","lastUpdate":1262210916932}})

In the next example the tab displaying the JSON web page has been closed and a new tab opened to my web site. It can be seen that the JSON site details are recorded after the "_closedTabs" name. Following that there is a list of hosts at "_hosts" and closed windows at "_closedWindows" which in this case is an empty array [].

({"windows":[{"tabs":[{"entries":[{"url":"http://computerforensics.parsonage.co.uk/","title":"Computer Forensics Miscellany","ID":2,"scroll":"0,0"}],"index":1,"attributes":{"image":"http://computerforensics.parsonage.co.uk/favicon.ico"},"pageStyle":"CSS","_formDataSaved":true}],"selected":1,"_closedTabs":[{"state":{"entries":[{"url":"http://www.json.org/","title":"JSON","ID":0,"scroll":"0,0"}],"index":1,"attributes":{"image":"http://www.json.org/favicon.gif"},"_formDataSaved":true},"title":"JSON","image":"http://www.json.org/favicon.gif","pos":0}],"_hosts":{"parsonage.co.uk":true,"co.uk":true,"uk":true,"computerforensics.parsonage.co.uk":true},"width":"1628","height":"996","screenX":"33","screenY":"24","sizemode":"maximized"}],"selectedWindow":1,"_closedWindows":[],"session":{"state":"running","lastUpdate":1262211186521}})

When more tabs are opened and closed and the pages being viewed are less basic than those in the examples above the file becomes increasingly complex. Viewing the file in a suitable code editor will assist by formatting the file into a more readable structure.

Recovering Session Restore files

It is a trivial process to recover Session Restore files. I used the Case Processor File Finder script in Encase and added a Custom File Type using **({"windows":[{** as the header and **:{"state":"running"}})** as the footer. (NOTE: In the case I was examining from 2008 there was no "lastUpdate" name/value pair. It may require some experimentation to find the exact structure in your case as these browsers are constantly being developed). There is also an alternative **"state"** string of **"stopped"** which I used in a second Custom File Type as the footer with the same header of **({"windows":[{**, however I found no instances of this in unallocated and the only two live files contained no windows and tab data.

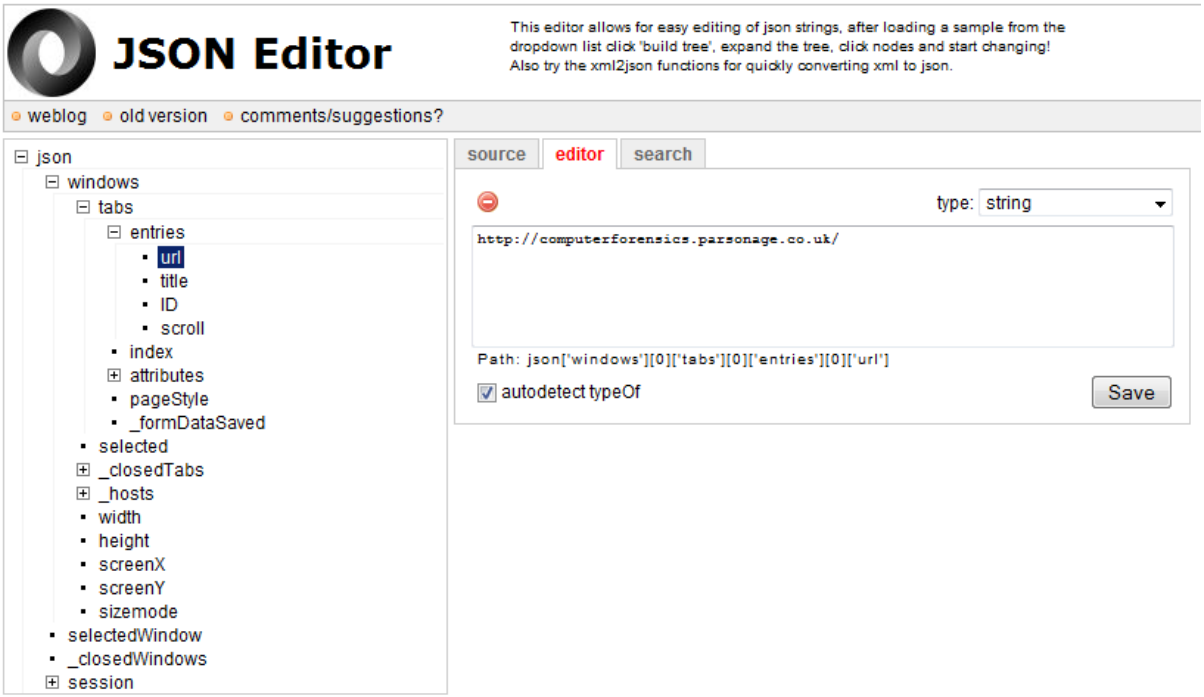
You will need to use the escape character "\" in the search string to escape the special characters otherwise your search will not work.

I exported all the recovered Session Restore files as well as bookmarking them.

Examining the Session Restore Files

Depending upon how many files are recovered, you might want to reduce the number of files by using text strings searches to identify those files that might be of interest.

There is an online JSON Editor (3) where you can paste the contents of a ***sessionstore.js*** file and it will build a tree structure displaying the values.



Allan Hay as also made available his JSON editor program, a single executable which can be used offline. (4)

It would be a tedious process to examine many recovered Session Restore files using an editor so an alternative method would be to use the Mozilla browser itself.

I did this using a Session Manager Add-In (5) as it had more functionality than Firefox alone.

The process I used was this –

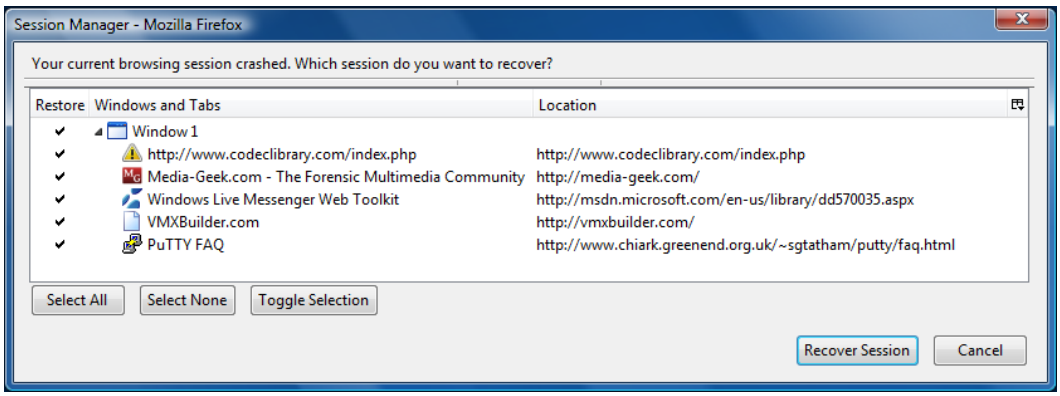
1) Have the folder containing the carved Session Restore files open.

2) Have the C:\Users\UserName\AppData\Roaming\Mozilla\Firefox\Profiles\########.default folder open.

3) Clear History and Close Firefox.

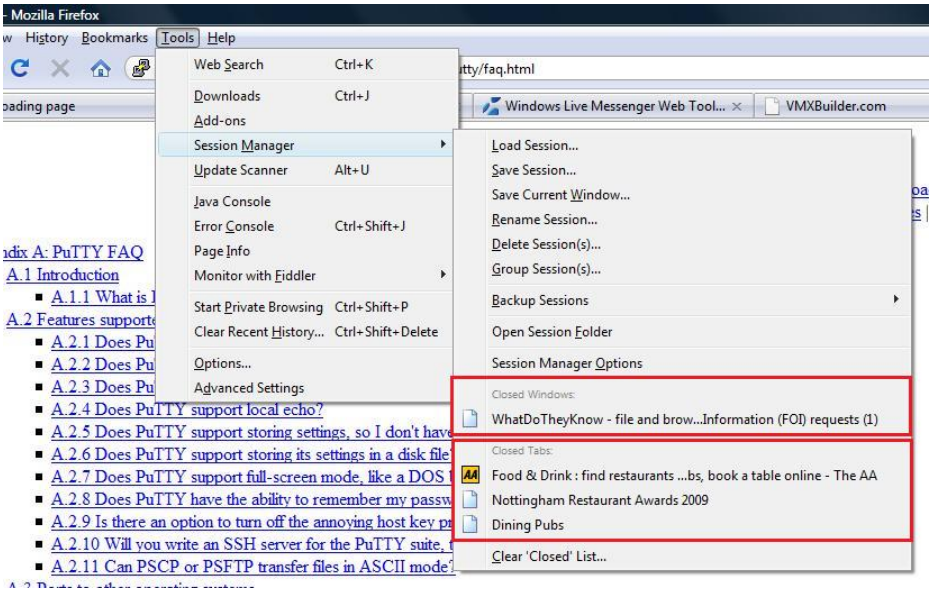
4) Drag and drop a copy of the first carved file into the \########.default folder and rename it to ***sessionstore.js***

5) Open Firefox and the Session Manager Add-In will recognise that there is a crashed session available to restore. The manager will list the number of windows and tabs in the session and also each web page title and url. Take a screenshot of the Session Manager to record the details.



6) If required you could restore the session to the browser and see the detail of the web pages if they currently exist with a live Internet connection. The pages will even be scrolled to the location that was previously being viewed and the windows will be sized and located as they were at the time of the session.

7) The Closed Tabs and Closed Windows (if there are any) can be identified via the Tools\Session Manager menu item.



8) If the "lastUpdate" name/value pair is available in the Session Restore file this can be decoded using TimeLord (6) to give the time and date of the session.

Mark Woan has also developed an application to parse out the main parts (but not all) of the ***sessionstore.js*** file and present it in a more readable format. (7) It can be used to parse all the files in a single folder when a number of ***sessionstore.js*** files have been carved from a case. Using Mark's application it is possible to pipe all the results into a single file for easier reviewing.

As well as providing a snapshot in time of a user's windows and tabs a ***sessionstore.js*** file can also provide a history of browsing. Where a user has had a single tab open and then browsed to a number of sites in that tab, each site visited is recorded in a list under that tab in the order in which the sites were accessed. It is not known if there is a limit to the number of sites that can be listed but I have observed in one test that there was a list of all of the 22 sites I had visited under one tab. Just note that the Session Manager does not display this information but Mark's parser does.

When parsed by Mark's application the start of the list looked like this –

Entries
---
title : Computer Forensics Miscellany
url : http://computerforensics.parsonage.co.uk/

title : X-Ways Support Forum
url : http://www.winhex.net/

title : Telegraph Fantasy Football 2009/10 = Fantasy Football
url : http://fantasyfootball.telegraph.co.uk/

If you have found any sessions that are of interest you may then return to the JSON Editor and the raw ***sessionstore.js*** file itself to examine it carefully so you can be satisfied that the Session Manager or other tool is displaying what you would expect and so that you can explain what is being displayed should this become necessary.

Further opportunities for research in this area – Internet Explorer 8

A brief examination of Internet Explorer 8 reveals that there is a similar Session Restore feature available in this browser, which I believe was also available in Version 7 too.

The files relating to the feature are stored in –

Windows XP

C:\Documents and Settings\UserName\Local Settings\Application Data\Microsoft\Internet Explorer\Recovery\Last Active

Windows Vista and Windows 7 in Normal Privilege Mode

C:\Users\UserName\AppData\Local\Microsoft\Internet Explorer\Recovery\Last Active

Windows Vista and Windows 7 in Elevated Administrator Privilege Mode

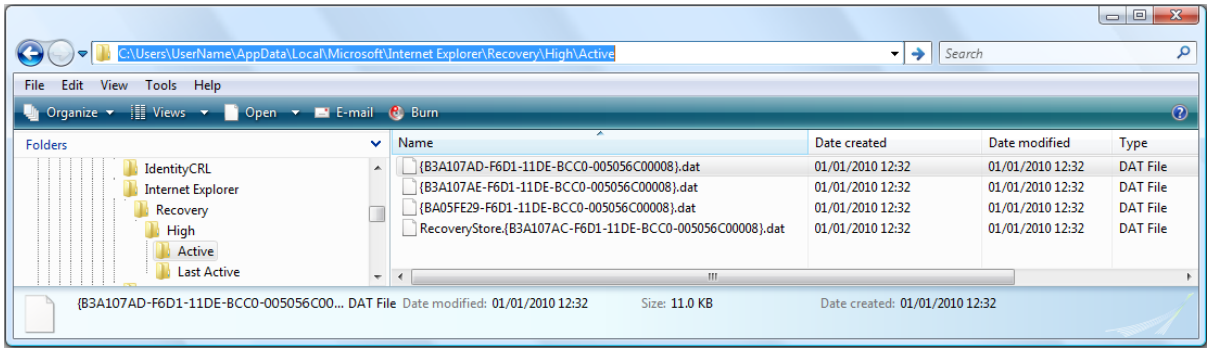
C:\Users\UserName\AppData\Local\Microsoft\Internet Explorer\Recovery\High\Last Active

Whilst the browser session is active the files are stored in a folder named “Active” on the same level as “Last Active” and when the browser closes normally they are moved from “Active” to “Last Active”.

The Session restore files are named in the form –

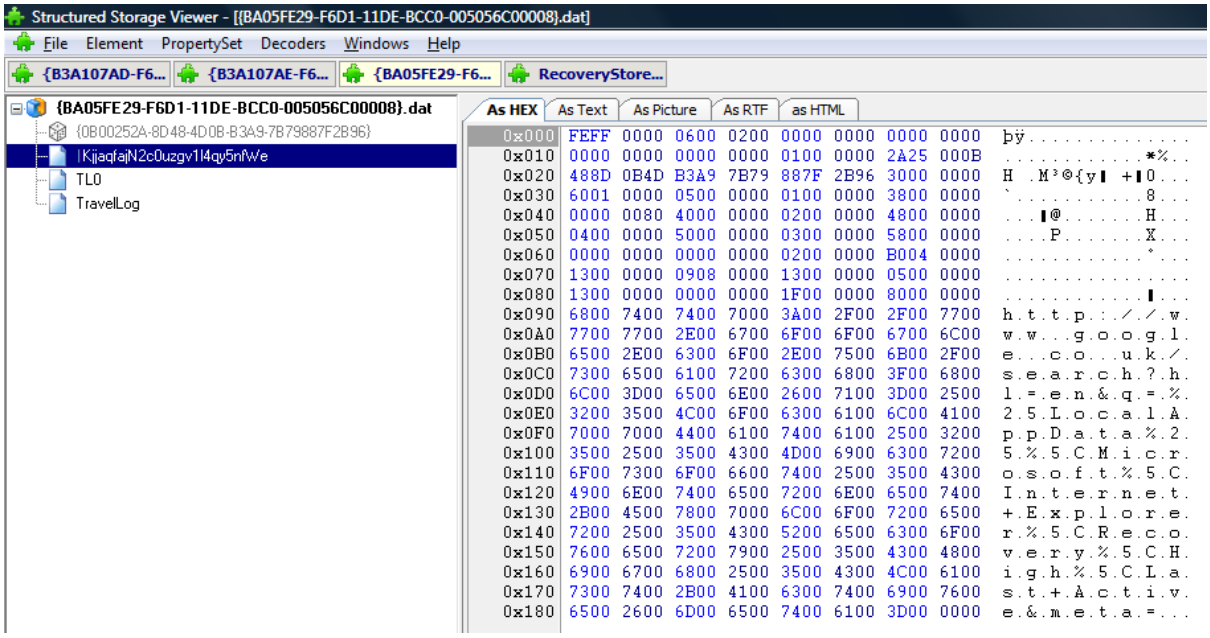
{GUID}.dat and

RecoveryStore. {GUID}.dat

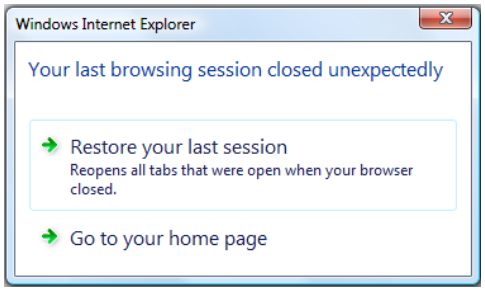


The files are stored in Microsoft’s Compound File Binary File Format (8) and the RecoveryStore. {GUID}.dat file references each of the individual {GUID}.dat files. The individual {GUID}.dat files

appear to contain the details of each tab but are in binary format and so not as readily understood as the Mozilla equivalents. The screenshot below shows one of the data streams in a {GUID}.dat file.

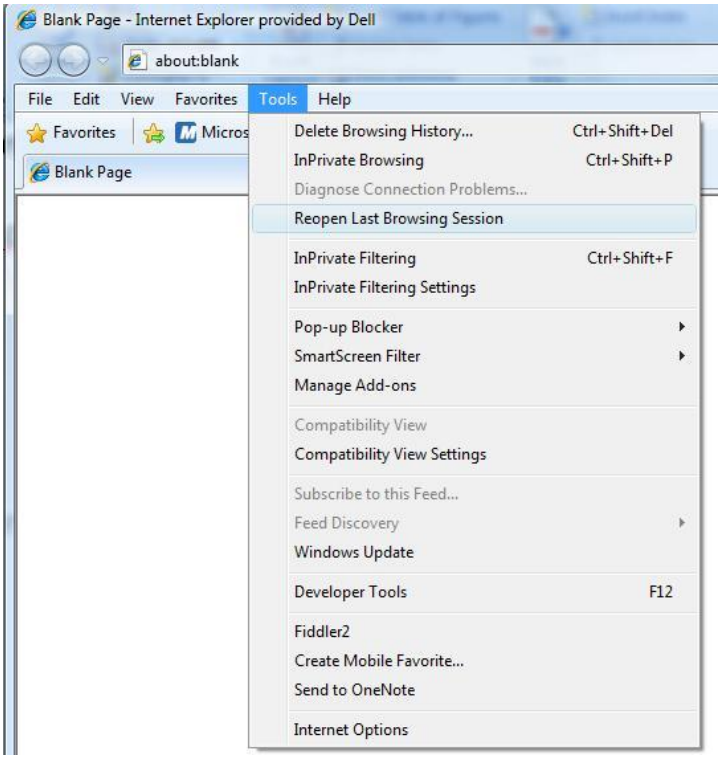


When the user has suffered a browser crash the next time Internet Explorer is opened the user is given the following option.



Choosing “restore your last session” does just that.

If the browser has closed normally the user has the option to “Reopen Last Browsing Session”.

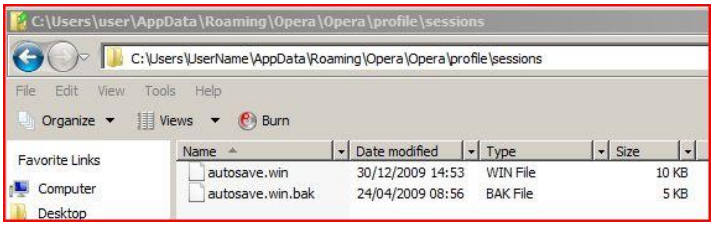


It should be possible to examine a user's previous browser session by copying the recovery files from the suspect machine and placing them in the appropriate folder on an examiner's machine. Alternatively this could be done using the suspect computer running as a Virtual Machine.

In principle it may be possible to search for compound files in unallocated space and carry out the same process as suggested for Mozilla but the likelihood of success will be considerably reduced. It will no doubt be possible to recover artefacts of the {GUID}.dat files and draw some conclusions from these.

Opera Browser

I checked the Opera browser and found this too had a Session Restore facility stored in a raw text file - ***autosave.win***, with a backup file ***autosave.win.bak***.



The both files are located in the same folder –

XP

C:\Documents and Settings\UserName\Application Data\Opera\Opera\sessions

Vista

C:\Users\ UserName \AppData\Roaming\Opera\Opera\profile\sessions

The files contain similar details to the Mozilla files and are much easier to read.

The file looks like this –

```
Opera Preferences version 2.1
; Do not edit this file while Opera is running
; This file is stored in UTF-8 encoding

[session]
version=7000
window count=4

[1]
x=100
y=100
w=700
h=500
state=2
restore to state=2
id=173
parent=0

------/ snip / ------

[2history url]
count=1
0=http://www.amazon.co.uk/?ie=UTF8&%2AVersion%2A=1&tag=operasoft-
21&link_code=hom&%2Aentries%2A=0

[2history document type]
count=1
0=6

[2history title]
count=1
0=Amazon.co.uk: Low Prices in Electronics, Books, Sports Equipment & more

------/ snip / ------
```

```
[4history url]
count=1
0=http://computerforensics.parsonage.co.uk/

[4history document type]
count=1
0=6

[4history title]
count=1
0=Computer Forensics Miscellany

[4history scrollpos list]
count=1
0=0
```

Safari v4

Following my initial version of this paper Dc 1743 has added some Apple Mac related information on his blog (9) -

For Safari v4 the last session information is contained in a file entitled LastSession.plist

In Mac OSX 10.6 this file is stored at /Users/<user name>/Library/Safari

In XP this file is stored at C:\Documents and Settings\<User name>\Application Data\Apple Computer\Safari

I use the mac application - property list editor to review plists, there are windows applications to do this as well.

Firefox v3.5.6 running in Mac OSX 10.6

The sessionstore.js file is stored at /Users/<User Name>/Library/Application Support/Firefox/Profiles/XXXXXXX.default

If anyone has any comments, suggestions or updates regarding this topic please contact me by email.

I can be contacted at digitalforensics@ my domain.

Bibliography

1. Session Restore. *MozillaWiki.* [Online] [Cited: 1 January 2010.] https://wiki.mozilla.org/Session_Restore.

2. JavaScript Object Notation. *JSON.* [Online] [Cited: 1 January 2010.] http://www.json.org/.

3. *JSON Editor.* [Online] [Cited: 1 January 2010.] http://braincast.nl/samples/jsoneditor/.

4. **Hay, Allan.** JSON Editor Program. [Online] [Cited: 8 January 2010.] http://computerforensics.parsonage.co.uk/downloads/JSONv1.1.zip.

5. *mozdev - Session Manager.* [Online] [Cited: 1 January 2010.] http://sessionmanager.mozdev.org/.

6. TimeLord by Paul Tew - A Time Utility for Computer Forensic Analysts. *Computer Forensic Miscellany.* [Online] Parsonage Computer Forensics. [Cited: 1 January 2010.] http://computerforensics.parsonage.co.uk/timelord/timelord.htm.

7. **Woan, Mark.** woanware. [Online] [Cited: 8 January 2010.] http://www.woanware.co.uk/firefoxsessionstoreextractor/.

8. **Microsoft.** [MS-CFB] Compound File Binary File Format. *Microsoft Downloads.* [Online] [Cited: 1 January 2010.] http://download.microsoft.com/download/1/6/f/16f4e321-aa6b-4fa3-8ad3-e94c895a3c97/%5BMS-CFB%5D.pdf.

9. **Dc1743.** *Forensics from the sausage factory.* [Online] [Cited: 05 January 2010.] http://forensicsfromthesausagefactory.blogspot.com/.