Chapter 9

FORENSIC ANALYSIS OF THE SONY PLAYSTATION PORTABLE

Scott Conrad, Carlos Rodriguez, Chris Marberry and Philip Craiger

Abstract The Sony PlayStation Portable (PSP) is a popular portable gaming device with features such as wireless Internet access and image, music and movie playback. As with most systems built around a processor and storage, the PSP can be used for purposes other than it was originally intended – legal as well as illegal. This paper discusses the features of the PSP browser and suggests best practices for extracting digital evidence.

Keywords: Sony PlayStation Portable, forensic analysis

1. Introduction

The Sony PlayStation Portable (PSP) is a popular portable video game system that has additional multimedia and Internet-related capabilities. Originally released in 2004, the PSP features a 4.3" widescreen LCD with 480×272 pixel resolution. It comes with a dual core 222 MHz R4000 CPU, 32 MB RAM and 4 MB of embedded DRAM, which holds the operating system [6]. The PSP uses a proprietary Universal Media Disk (UMD) as its primary read-only storage media for games and movies. The device also features 802.11b Wi-Fi connectivity for multiplayer games and utilizes a Pro Duo memory stick for secondary storage.

In September 2007, Sony released a new version of the PSP that is 33% lighter and 19% thinner, appropriately dubbed the PSP Slim & Lite. The Slim & Lite version caches UMD data in memory to decrease game loading time and provides additional features such as a brighter screen, composite TV output, charging via USB and double the onboard RAM (64 MB) [8].

The PSP has updatable firmware that can be downloaded from a Sony server using the Wi-Fi interface. Version 2.0 of the firmware provides a

browser and a Really Simple Syndication (RSS) reader. The RSS reader can connect and pull in content via RSS "feeds" typically provided by rapidly updated websites that can be viewed outside of a web browser. The content includes blog entries, news headlines, audio and video. RSS allows for subscriptions to favored content and aggregated feeds from multiple sites [3]. Because these feeds are completely user-defined, they can provide considerable information about the browsing habits of users.

Sony selected the NetFront browser from Access as the internal web browser for the PSP. NetFront is currently deployed in more than 139 devices, including mobile phones, PDAs, digital TVs, gaming consoles and automobile telematics systems from 90 major Internet device manufacturers [10]. The browser has robust capabilities via features such as HTML 4.01 support, flash support, CSS support, tabbed browsing, offline browsing, SSL support, streaming downloads and Smart-Fit rendering [5]. In addition, NetFront provides features associated with traditional web browsers, including the ability to save bookmarks and URL history, both of which provide additional information about the browsing habits of users.

This paper examines the principal features of the PSP browser, in particular, the data structures used to save bookmarks, URL history and other information about user browsing habits. It also presents forensically-sound techniques that can be used to extract digital evidence from the Sony PSP.

2. Background

In April 2005, a DNS redirection flaw was discovered in the content downloading feature of the *Wipeout Pure* video game that enables web pages other than the official game website to be displayed. This discovery drew attention to the fact that addresses such as `file:///disc0:/` enable UMD files to be viewed; these files are normally hidden from users. Soon after the discovery, a method for formatting PSP executables (e.g., `EBOOT.BIN`) was devised, which brought the inner workings of the PSP to light. (The `EBOOT` file is a packaged executable file, much like a traditional `.exe` file.) Some time later, a hacker named "NEM" and the "Saturn Expedition Committee" were able to successfully reverse engineer the layout of the executable format [9].

These exploits and others enable programmers to modify the firmware directly, allowing unsigned software and third party ("homebrew") applications to be run on the PSP; this is possible because the PSP lacks a mechanism to verify that executables are digitally signed. It is also possible to execute third party applications from a memory stick by mod-

ifying Version 1.00 of the Sony firmware. In fact, every version of the PSP firmware has been modified and countless homebrew applications have been developed for the PSP.

Homebrew applications are not always designed for illicit purposes, although some exist solely to circumvent copyright protection. Quite often, they are a way for independent developers to demonstrate their creativity by creating their own PSP games.

3. Memory

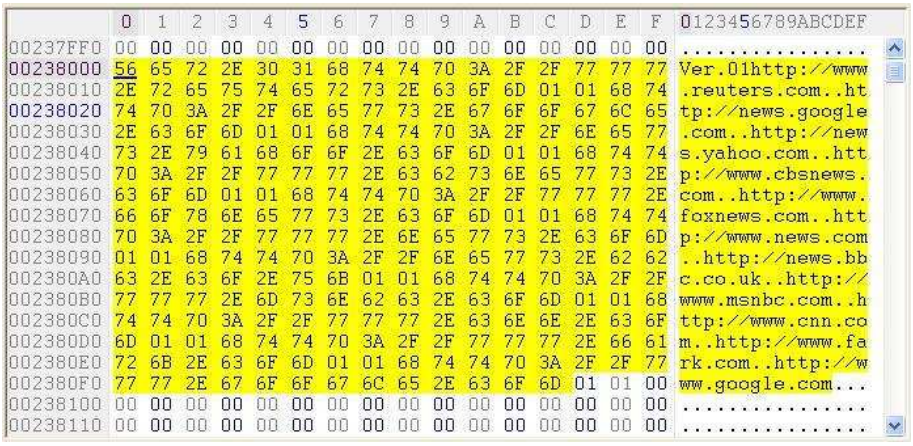
The PSP memory stick has a FAT16 file system. Thus, standard forensic software, such as Encase, FTK and hex editors, can be used to analyze the memory stick. The memory stick used in our tests was 1 GB in size; the cluster size in the FAT16 file system was 32 KB.

The PSP also has a significant amount of RAM or cache memory. However, there is no way to directly access the memory and copy the contents other than to physically remove chips from the PSP (which can be extremely risky). Custom firmware is available to obtain a memory dump, but the techniques may not be forensically sound. Also they present a *Catch 22* situation: the only practical way to extract data from PSP RAM is to install software that overwrites some of the RAM data. However, this is not a serious problem because PSP RAM does not hold important user data; it almost exclusively stores firmware and various system settings. In fact, the only user-generated data stored in RAM is the background picture, and only if the user has changed it from the default picture. For this reason, the rest of this paper focuses exclusively on data stored in the memory stick.

4. Browser History Files

The PSP web browser stores the browsing history in various files on the memory stick, each file reflecting a different aspect of the history. Most of the data in the history files is stored as plaintext and is thus easily searched. For example, a search for "`http:`" will almost always find at least the `historyv.dat` file and searches of the data within `historyv.dat` will usually find the other history files (`historyi.dat` and `historys.dat`) if they exist on the memory stick.

Testing revealed that the file system usually begins the next history file two clusters below the beginning of the previous history file, i.e., if `historyi.dat` begins at the relative hex offset of `170000`, then `historyv.dat` begins at the relative offset of `180000`. Also, the browser must be shut down gracefully for the history files to be written to the memory stick. This is because the browser does not constantly write to



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
00237FF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00238000	56	65	72	2E	30	31	68	74	74	70	3A	2F	2F	77	77	77	Ver.01http://www
00238010	2E	72	65	75	74	65	72	73	2E	63	6F	6D	01	01	68	74	.reuters.com..ht
00238020	74	70	3A	2F	2F	6E	65	77	73	2E	67	6F	6F	67	6C	65	tp://news.google
00238030	2E	63	6F	6D	01	01	68	74	74	70	3A	2F	2F	6E	65	77	.com..http://new
00238040	73	2E	79	61	68	6F	6F	2E	63	6F	6D	01	01	68	74	74	s.yahoo.com..htt
00238050	70	3A	2F	2F	77	77	77	2E	63	62	73	6E	65	77	73	2E	p://www.cbsnews.
00238060	63	6F	6D	01	01	68	74	74	70	3A	2F	2F	77	77	77	2E	com..http://www.
00238070	66	6F	78	6E	65	77	73	2E	63	6F	6D	01	01	68	74	74	foxnews.com..htt
00238080	70	3A	2F	2F	77	77	77	2E	6E	65	77	73	2E	63	6F	6D	p://www.news.com
00238090	01	01	68	74	74	70	3A	2F	2F	6E	65	77	73	2E	62	62	..http://news.bb
002380A0	63	2E	63	6F	2E	75	6B	01	01	68	74	74	70	3A	2F	2F	c.co.uk..http://
002380B0	77	77	77	2E	6D	73	6E	62	63	2E	63	6F	6D	01	01	68	www.msnbc.com..h
002380C0	74	74	70	3A	2F	2F	77	77	77	2E	63	6E	6E	2E	63	6F	ttp://www.cnn.co
002380D0	6D	01	01	68	74	74	70	3A	2F	2F	77	77	77	2E	66	61	m..http://www.fa
002380E0	72	6B	2E	63	6F	6D	01	01	68	74	74	70	3A	2F	2F	77	rk.com..http://w
002380F0	77	77	2E	67	6F	6F	67	6C	65	2E	63	6F	6D	01	01	00	ww.google.com...
00238100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00238110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Figure 1. Format of `historyi.dat` pages.

the memory stick while it is being used. Instead, the browser keeps everything in internal memory and writes the history files to the memory stick just before it closes. This means that if the PSP is turned off, or if the memory stick is removed before the web browser is exited, or even if the PSP is returned to the home page without closing the web browser, then the history files are not written to the memory stick.

The first history file, which stores all the manually-typed web addresses, is found in the following location on the memory stick:

```
X:\PSP\SYSTEM\BROWSER\historyi.dat
```

Note that `X:` is the drive letter assigned to the memory stick.

A sample page in `historyi.dat` is shown in Figure 1. The pages in the file have the format:

```
<version number>[typed address](white space)
                [typed address](white space)
                [typed address](...)
```

where the `version number` is usually "Ver.01."

The web addresses stored in `historyi.dat` are not necessarily those that were visited; they are merely those that the user manually typed into the browser and attempted to visit. Entries in the file appear exactly as they were typed, i.e., `http://` is not automatically added to the beginning of a typed URL. Additionally, the most recently typed addresses are placed at the beginning of `historyi.dat` instead of being appended to the end, and an entry only appears once in the file regardless of how many times it was typed into the browser. When an entry is repeated, it is simply moved to the beginning of the file.

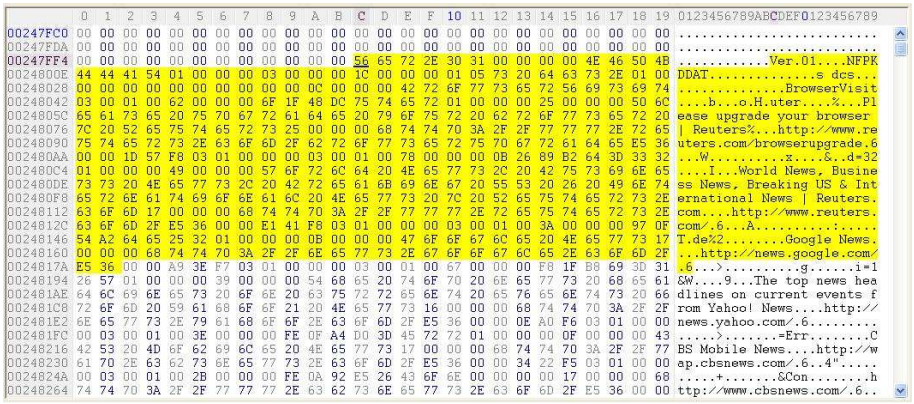


Figure 2. Format of `historyv.dat` entries.

The second history file stores web addresses that are actually visited, whether they are manually typed or accessed via html links:

`X:\PSP\SYSTEM\BROWSER\historyv.dat`

As with the `historyi.dat` file, the most recently visited web address appears at the beginning of the `historyv.dat` file. Unlike the `historyi.dat` file, entries can appear multiple times if they were accessed more than once. The URLs in `historyv.dat` are always valid web addresses. Also, the `historyv.dat` file can be transferred to the `\Browser` folder in any PSP memory stick and the PSP browser may be used to display the title, address and last accessed dates of all the entries in the file. This is very useful because the last accessed date is normally stored within the encoded data.

Figure 2 shows a portion of the `historyv.dat` file. The format of each entry in the file is:

```
<version number><encoded data>
      [website HTML title (if applicable)]
      [(URL protocol)(website address)]
      <_6><encoded data>
            [Website HTML title (if applicable)]
            [(URL protocol)(website address)]
      <_6><encoded data>(...)
```

where the `version number` is usually "Ver.01."

5. Internet Search Feature

Sony released Version 4.0 of the PSP firmware in June 2008. This firmware update enables users to perform Internet searches directly from

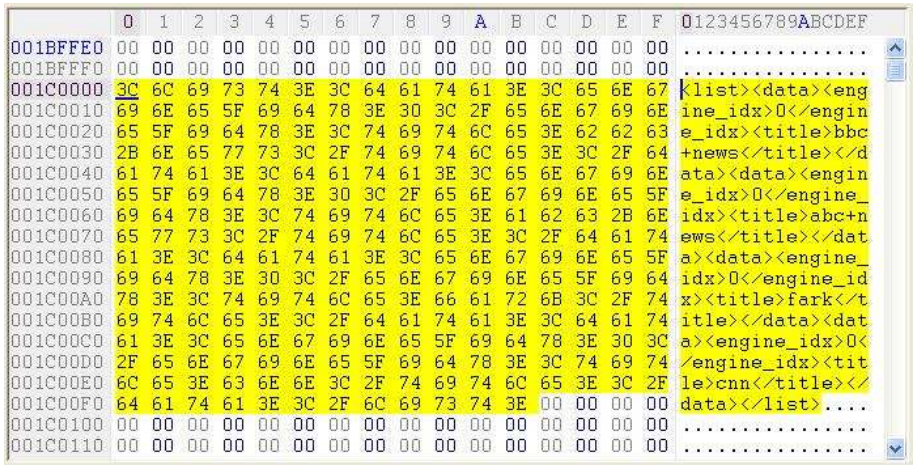


Figure 3. Format of `historys.dat` entries.

the PSP Home Menu [11]. It appears that Google is the default search engine.

The `historys.dat` file that stores the corresponding information is located at.

```
X:\PSP\SYSTEM\BROWSER\historys.dat
```

Figure 3 shows a portion of the `historys.dat` file. The format of file entries is:

```
<list><data><engine_idx>[generated number]
            </engine_idx><title>
                  [query]</title></data><data>
            <engine_idx>[generated number]
            </engine_idx><title>
                  [query]</title></data>...</list>
```

Note that the recording format is very similar to that of common markup languages such as HTML and XML. Unlike the `historyi.dat` and `historyv.dat` files, every time a new query is performed using the Internet Search feature, a new `historys.dat` file is created that shows the new query with the previous queries appended at the end.

6. RSS History

Version 2.6 of the PSP firmware (released in November 2005) added support for RSS feeds [11]. The RSS Channel feature is presented to users above the Web Browsing option. This mobile RSS aggregator was originally designed for downloading web feeds and pod casts in MP3 or

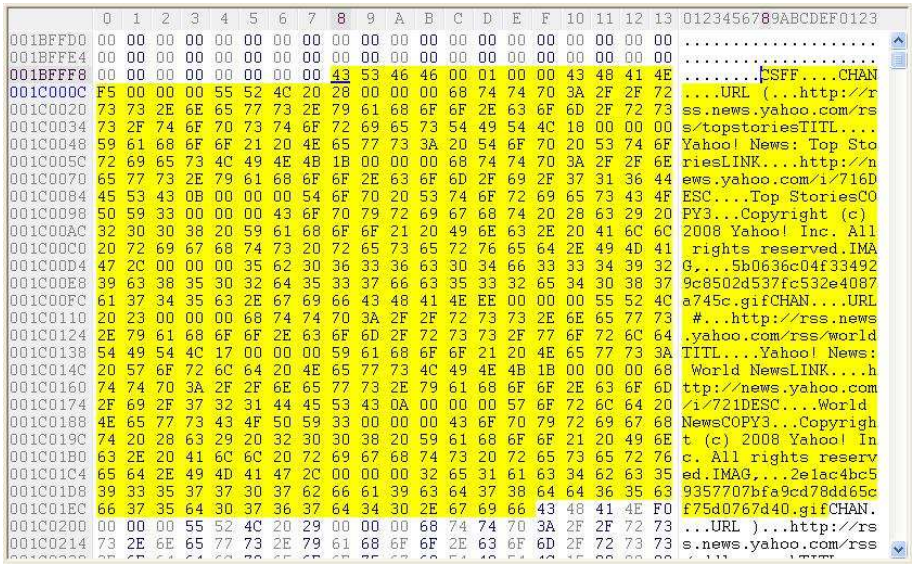


Figure 4. Format of `CHLIST` entries.

AAC formats. Version 2.8 added support for downloading video and image content [7]. All RSS content may be downloaded directly to a memory stick in the PSP. The data downloaded from a RSS feed is stored in the `CHLIST` file:

```
X:\PSP\SYSTEM\RSSCH\CHLIST
```

Figure 4 shows a portion of the `CHLIST` file. The format of file entries is:

```
CSFF<binary data>
CHAN<binary data>
URL<binary data>[URL of RSS feed (with http://)]
TITL<binary data>[Title of website]
LINK<binary data>[URL of website behind RSS feed]
DESC<binary data>[Description of website]
COPY<binary data>[Copyright info]
IMAG<binary data>[Name of associated image
                  (also saved in ...\RSSCH)]
CHAN<binary data>[etc.]
```

For each entry in the `CHLIST` file, there is a corresponding image linked to that entry in the same folder (`...\RSSCH`). Note that the topmost entries in the file are the oldest accessed RSS feeds and the entries towards the end of the file are the most recently accessed RSS feeds.

7. Persistence of Deleted History Data

The main objective of our research was to analyze the data structures used by the PSP web browser when storing web history. Our first step was to examine how deleted data behaved in the PSP. We formulated a test to enable us to discern how the PSP manipulates history data. The tools used to conduct this test included a Windows workstation (Windows XP), Hex Workshop (hex editor) [1], `dd` [4] (used for byte-level copying of raw data from the physical memory stick to an image file), a Tableau USB write blocker and a Sony PSP Slim & Lite with a 1 GB Pro Duo memory stick.

We used Hex Workshop to wipe the memory stick by writing `00` to every byte. The clean memory stick was then inserted into the PSP and formatted (`Settings>System Settings>Format>Memory Stick`). Next, the PSP browser was launched and several web addresses were visited in sequence after each website was allowed to load completely. The browser was exited gracefully and the memory stick was removed from the PSP and connected to a write blocker. A raw image of the memory stick was created using `dd` and saved as `Before.001`. The memory stick was then removed from the write blocker and connected to the workstation. The history files were manually deleted from the directory `X:\PSP\SYSTEM\BROWSER` using Windows Explorer. The memory stick was placed back in the PSP and several new web addresses were accessed via the PSP browser. The memory stick was then removed from the PSP and connected back to the write blocker and another raw image was created (`After1.001`). Next, `dd` was used to restore the image `Before.001` to the memory stick. The memory stick was placed back in the PSP, the browser was launched and the history was cleared completely using the following steps:

```
History>Options>Delete All
Tools>Delete Cookies
Tools>Delete Cache
Tools>Delete Authentication
Tools>Delete Input History
```

After the history was deleted, a second set of web addresses was visited using the browser. The memory stick was removed from the PSP, connected again to the write blocker and a raw image was created (`After2.001`). The three images, `Before.001`, `After1.001` and `After2.001`, were compared and analyzed using a hex editor.

After analyzing the files, it was discovered that when the history is erased from the PSP browser using the method mentioned above (image `After2.001`), the browser generally does not overwrite the old history

when it begins a new history. Instead, the file system (most of the time) simply moves down one cluster from the beginning of the old history file. In another words, if the old `historyi.dat` begins at relative hex offset `170000`, then the new `historyi.dat` begins at `178000`, which means that both the old `historyi.dat` and the old `historyv.dat` (which are usually located at an offset of `180000`) would not be overwritten.

In contrast, when the browser history is deleted using Windows Explorer (`After1.001`), the old history is generally overwritten by the new history. This is because the PSP does not move down a cluster before saving the new data. However, if the new history is smaller than the old history (i.e., if the old history has twenty entries and the new history only has ten entries), then parts of the old history are still recoverable.

We discovered that the only data that was consistently altered was the data located in the FAT. However, the first few bytes of each deleted history were almost always changed to indicate that they were deleted and not active. Depending on the circumstances, the actual history files were untouched, not entirely overwritten or completely unrecoverable. In general, however, the closer the history files are to the end of the memory stick, the longer they survive.

8. Persistence of Overwritten Data

Peculiar behavior was observed when performing the test described above to study the persistence of deleted history data. For some reason, overwritten data in the hex address range of `168000-1FFFFF` can be recovered completely by formatting the memory stick. This behavior was confirmed by connecting a memory stick directly to the workstation and completely filling it with `aa` values using a hex editor. After it was confirmed that the memory stick only had `aa` values written to it, it was completely filled again, but this time with `bb` values. The memory stick was then placed in the PSP and formatted using the built-in function. Finally, the memory stick was connected to a write blocker and a search for `aa` values was conducted; these were always the only values stored in the `168000-1FFFFF` address range. The rest of the memory stick was filled with `bb` values, except for the locations holding the file system. This test was conducted several times with values other than `aa` and `bb`. One test even used the 16-byte pattern `12 23 34 45 56 67 78 89 90 0a ab bc cd de ef f1` instead of `aa`. Nevertheless, the overwritten data always reappeared after being formatted.

This peculiar behavior does not appear to be due to some unknown function of the PSP; rather, it is due to the combination of some physical property of the memory stick and the way the format function works.

This was verified by placing a memory stick in a PSP, writing history files to it and then having the new history files overwrite the old files. The memory stick was then formatted on a completely different PSP and the same behavior was observed. Since it was not possible for the second PSP to rewrite the history files that were created by first PSP, it is apparent that the behavior is not caused by the PSP directly; instead, it has something to do with a physical property of the memory stick. Finally, the fact that the behavior was not observed when the memory stick was formatted by the Windows workstation shows that the PSP formatting function is involved rather than a specific PSP device.

9. Conclusions

The Sony PSP is not merely a portable gaming console, but a sophisticated device with considerable storage capacity and Internet access. Indeed, it provides the functionality of a small personal computer.

Our research demonstrates that it is possible to recover web browsing history and RSS subscription information from a PSP. Several methods have been proposed for identifying and recovering this information. But further research is required to examine the forensic implications of other PSP features, especially as Sony continues to develop PSPs with new functionality.

Digital forensic investigators are certain to encounter increasing numbers of Sony PSPs and other gaming devices in their crime scene investigations. A modified Xbox [2] is capable of running Linux applications; other game devices can run Linux without any modifications. Consequently, it is important that digital forensic researchers focus on gaming devices, conduct comprehensive examinations of their advanced features, determine the locations where evidence may reside, and develop forensically-sound methodologies for recovering the evidence.

References

[1] BreakPoint Software, Hex Workshop, Cambridge, Massachusetts (www.hexworkshop.com).

[2] P. Burke and P. Craiger, Forensic analysis of Xbox consoles, in *Advances in Digital Forensics III*, P. Craiger and S. Shenoi (Eds.), Springer, Boston, Massachusetts, pp. 269–280, 2007.

[3] R. Cadenhead, G. Smith, J. Hanna and B. Kearney, The application/rss+xml media type, Network Working Group (www.rssboard .org/rss-mime-type-application.txt), 2006.

[4] Free Software Foundation, `dd`: Convert and copy file, GNU Coreutils, Boston, Massachusetts (www.gnu.org/software/coreutils/manual/html_node/dd-invocation.html).

[5] J. Puente, What browser does the Sony PSP use? (jefte.net/psp/what-browser-does-the-sony-psp-use), 2006.

[6] J. Sanches, PSP Slim & Lite, Steel Media, Uxbridge, United Kingdom (www.pocketgamer.co.uk/r/PSP/PSP+Slim+&+Lite/hardware_review.asp?c=4188), September 18, 2007.

[7] Sony Computer Entertainment America, RSS document specifications, Culver City, California (www.playstation.com/manual/psp/rss/en/spec.html), 2007.

[8] Sony Computer Entertainment America, Sony Computer Entertainment America to offer limited-edition entertainment packs with newly designed PSP (PlayStation Portable) starting this fall, Culver City, California (www.us.playstation.com/News/PressReleases/407), July 11, 2007.

[9] Wikipedia, PlayStation Portable homebrew, Wikipedia Foundation, San Francisco, California (en.wikipedia.org/wiki/Psp_homebrew).

[10] WindowsForDevices.com, Access NetFront browser wins best software award, New York (www.windowsfordevices.com/news/NS7911853350.html), November 8, 2004.

[11] Xtreme PSP, Firmware history and compatibility (www.xtremepsp.com/firmware.php).