The 2012 **DFRWS** Data Sniffing Challenge

Submission deadline: **Jun 30, 2012**

The overall goal of this challenge is to bring the state of the art into the digital forensic practice by providing an open public venue for a best-of-breed competition.

We challenge the competitors to develop the fastest and most accurate data block classifier. The scoring will be based on the weighted scores of three criteria:

1. *Correctness*, as measured by *precision & recall rates*: 55%.
2. *Processing speed*, in terms of throughput & scalability: 30%.
3. *Quality of code* and multi-platform support: 15%.

Rules:

- You may enter individually, or as a team with no restrictions.
- Your tool must have a command line interface and work on *at least* one of the three main OS platforms—MS Windows, MacOS, Linux—preferably more. It can be implemented in any widely and freely available language platform.
- The tool must have a corresponding library/API such that it could be incorporated as part of other tools.
- Source code must be openly available under a free software license, such as those listed at http://www.gnu.org/licenses/license-list.html. The author(s) retain rights to the source code.
- You may incorporate third party free software, as long as it is compatible with your license and it is included with your submission. However, your submission will be judged on the contribution your own work brings to the challenge.
- Your submission must include clear instructions for building the tool from source code along with all relevant dependencies.
- DFRWS will publish the results of the Challenge, both in detailed and summary form, along with the methodology used and the source of the specific version of each tool.

Technical Requirements:

- Command line invocation:

```
$ <tool_name> <target> <block_size> [<concurrency_factor>]
```

- Tools *must work right out of the box*, and will be tested both on actual drive images, as well as sequences of block samples glued together for convenience.
    - Whenever drive images are used, those will produced by repeated cycles of create/delete file operations; in other words, they will be realistic but of the "difficult" variety. Also, they may lack in certain details, such as filesystem metadata.
    - The target can be of substantial size, e.g. 100GB.

    - The target's file system could be any of FAT, NTFS, or ext3.
- The block sizes we will be testing for are 512, 1460, and 4096.
- The concurrency factor is optional. If your tool does support multi-threading/-processing, it will be tested with up to five values: 1, 4, 8, 12, 24 to evaluate its scalability on commodity hardware.

- **Output rules:**
    - The output should consist of one line per block and should identify the offset of the block and the type of data being detected by the tool.
    - If multiple types are detected, they should be outputted separated by space. The **first** type should identify the **top** level container (e.g., doc, pdf, etc.).
    - If your data sniffer is able to analyze popular encodings and identify the underlying data, it should first output the type of the data encoding (e.g. *base64*) and then the type of the underlying data (e.g., *jpg*), and connect the two by hyphen: e.g., *base64-jpg*.

    Example output:

```
> data_sniffer target 512
0 jpg                   JPEG data
512 jpg xml             XML inside a JPEG (presence of JPEG data is implied)
1024 jpg jpg            JPEG inside another JPEG (thumbnail)
1536 pdf jpg zlib       JPEG & deflate-compressed data as part of a PDF document
2048 html js            JavaScript inside html
2550 zlib-xml           Zlib-compressed xml
3092 pdf base85-jpg     PDF document with base85-encoded JPEG
3604 null               Unknown/unable to classify
```

Data types of interest:

The following is a list of the expected output file types and their respective interpretation. A tool's ability to handle additional common data types would be used to help decide a tie or near-tie.

`txt, csv, log`
: Text content: plain text, comma-separated values, system log. Note that the *csv* designation also covers the case where the fields are separated by a different character (<space>, <tab>, "|", etc.).

`html, xml, css`
: Web mark-up data: HTML-/XML-encoded data; CSS.

`js, json`
: JavaScript code, JSON data.

`base64, base85, hex`
: Text-encoded binary data: base64/85, hexadecimal encoded data.

`jpg, png, gif, fax, jbig`

Full-color image data: JPEG, PNG, GIF; bi-tonal images (common in scanned documents): CCITT Fax and JBIG.

`zlib, bz2`
General-purpose compression: DEFLATE (RFC 1951) and bzip2 (http://bzip.org).

`pdf`
Portable document format documents.

`ms-doc, ms-xls, ms-ppt`
Microsoft Office 97-2003 compound documents.

`ms-docx, ms-xlsx, ms-pptx`
Microsoft Office 2007 compound documents.

`mp3, aac`
Audio: MPEG layer III, AAC-encoded audio.

`h264, avi, wmv, flv`
Video encoding & packaging: H.264, AVI, WMV, Flash video.

`fs-fat, fs-ntfs, fs-ext`
Filesystem metadata for FAT, NTFS, ext3.

`encrypted, random, constant, null`
Special cases: encrypted, random, constant data, and unknown data. For the *constant* designation, at least half the block must be of the same value; constants may be 16 bits.

Clarifications (in response to questions submitted):

- Some of the classification tasks, such as distinguishing encrypted from random, are known to be difficult and may not be solvable in the general case.

- The dash notation indicates the underlying encoding of the same piece of data; space indicates separate, non-overlapping pieces. In the above example:

  `pdf jpg zlib` : this is part of a PDF doc, it contains (pieces of) a JPG-encoded element (streams in PDF parlance) and (pieces of) zlib-compressed data (probably text).

  `html js` : similarly: some html and some javascript detected.

  `zlib-xml` : this is zlib-compressed XML data (e.g. ms-docx); zlib would also be correct classification, but less specific

  `pdf base85-jpg` : this part of a PDF, it contains JPEG data that is base85-encoded; base85 would be correct but less specific

- Your tool may produce additional information/comment for a sample following a '#' sign. For example:

  ```
  1024 jpg #beginning of file
  ```

This is *strictly optional* and will only be taken into account to break ties.

Most test data will be obtained from public Internet sources. We expect that text content will be in English, however, no special filter will be applied. If you wish to obtain test data for development and tool testing, you may consider the data sets at http://digitalcorpora.org and http://www.cfreds.nist.gov, among other publically available.

Submission:

All participants must send an email to `challenge2012@dfrws.org` with the subject line "Solution submission". The email should contain official contact information for the participant/team members; it should also indicate to whom a check should be made out, in case the solution is selected for the grand prize.

The actual solution (code and relevant documentation) can be submitted in one of three ways:

- Email attachment. If the *entire* submission can be packed in an archive of less than 5MB, then submission can be sent as an attachment to `challenge2012@dfrws.org`.

- `http/ftp` download. The submission email can contain a download link from where the submission can be downloaded ***as a single file***.

- `svn/git` checkout. The submission email should contain appropriate instructions and credentials (if applicable) for organizers to obtain the submission.

Ideally, submissions should be self-contained; however, if bundling of third-party code is not possible (e.g., due to licensing restrictions) appropriate instructions on building the tool should be included.

As stated above, this competition is for open-source tools and, in the interest of open competition, DFRWS may publish the actual submissions along with test results. Beyond that, DFRWS will make no further attempts to distribute the solutions.

Although we strongly encourage toolmakers to cover as wide a range of data types as possible, all submissions will be given a fair chance, even if they do not cover all targeted data types.

Prizes:

First prize: DFRWS will provide free conference registration to our 2012 conference for up to two members of the winning team.

Grand prize: DFRWS seeks to award an additional $1,000 cash prize to the winners, if their solution exhibits all the attributes of a field-ready tool with the necessary robustness and performance.

The decision on prizes will be made by the DFRWS Organizing Committee based on the tool testing results conducted by the challenge team consisting of the following OC members: *Vassil Roussev*, *Wietse Venema*, and *Eoghan Casey*.

Contact:

Send all questions to `challenge2012@dfrws.org`. (Your email will be used *only* for this purpose and will be forgotten after DFRWS'12.)

Good luck!