Forensic Science International: Digital Investigation 32 (2020) 300910

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2020 EU — Proceedings of the Seventh Annual DFRWS Europe

PNG Data Detector for DECA

Kingson Chinedu Odogwu [a,*], Pavel Gladyshev [b], Babak Habibnia [b]. [a] *UCD School of Computer Science, University College Dublin, Belfied, Dublin 4, Ireland;* [b] *Digital Forensics Investigation Research Laboratory, University College Dublin, Belfied, Dublin 4, Ireland*

Keywords
File type classification
File type detection
Decision-theoretic file carving
File carving
Digital forensic investigation

A B S T R A C T

DECA, a novel file carving application, is an example of digital forensic tools that rely heavily on accurately detecting the type of data fragments stored in the disk blocks. This work is an attempt to create a method of detection and classification of PNG data types for DECA which originally only identifies and extracts JPEG data. The PNG file format was examined in order to implement the PNG data detector that was integrated into DECA. We then examined the results of decision-theoretic file carving, implemented in DECA, combined with the implemented PNG data detector.

* Corresponding author.

E-mail address: kingson.odogwu@ucdconnect.ie (K.C. Odogwu).

File fragment classification enables the collection and recovery of data and files by digital forensic tools, especially file carvers. DECA (decision-theoretic carving) program by Gladyshev and James (2017) is one of such file carvers. DECA originally implemented a JPEG data detector that makes use of escape sequences that appear in JPEG data. We decided to create a PNG data detector for DECA.

To create a PNG data detector for DECA, we looked into the PNG file format specification and PNG data files. We searched for a unique and consistent characteristic or feature that shows up in a lot of places in the PNG datastream, just like the escape sequence feature of JPEG data files, or that allows us to design a means of uniquely classifying or detecting PNG data fragments in data blocks.

From (Adler et al., 1999) and (Adler et al., 2017) we see that the unique byte values found in the PNG datastream are PNG signature, chunk type fields, and the IEND trailer. While these values do not occur frequently and consistently, it seemed the best way to identify PNG file fragments in memory.

We created a simple PNG data detector for experimentation that relies on the PNG signature and the byte representation of the chunk type field value of the critical chunks, ancillary chunks, and special-purpose chunks within the PNG file's datastream structure. The detector works with three different components. One component looks for the existence of the byte representation of chunk type field value (see Algorithm 1), another looks for the PNG signature and the other looks for the IEND trailer byte values.

Algorithm 1. Simple detector of PNG data

INPUT: array z[N] of 8-bit N bytes
RETURNS: 1, if z contains chunk type field values, otherwise return 0

```
// Iterate through the content of z and look to
// match with a chunk type field value
for (i = 0; i < index of last element in z; i++) {
  if (z[i] && the next 3 elements in z == chunk type field value) {
    return 1;
  }
}
return 0;
```

When DECA is operating in its sampling mode, it uses the PNG data detector to verify if a data block may contain the starting point of a PNG chunk. If successfully discovered, DECA switches to its actual file carving process, where it goes back to the block just after the previous block it inspected, and then it switches into its linear carving mode to identify the PNG signature and start extracting the PNG data till it gets to the IEND trailer.

The described algorithm for PNG data detector was implemented using the C programming language. It is now stored in DECA's code repository (Gladyshev, et al., 2016) after been integrated into DECA's source code.

As part of the development of the PNG data detector, experiments were designed to determine how well the PNG data detector integrated into DECA would perform. We used disks images from NIST's Computer Forensic Reference Data Sets (CFReDS), retrieved from the computer forensic tool testing (CFTT) section of the website (National Institute of Standards and Technology, 2019), and disk images produced from an

https://doi.org/10.1016/j.fsidi.2020.300910

alleged recruit's computer hard-disk, from a simulated terrorist recruitment activity exercise. DECA's code repository (Gladyshev, et al., 2016) now has this disk image and the image files stored in it.

Points of interest from these experiments were the number of PNG files extracted in the quickest processing time while looking out for the amount of false-positives or false-negatives in the output. Photorec (version 7.1) (CGSecurity, 2019), operating in its default settings asides specifying only PNG files to be extracted, was used to retrieve PNG image files from these disk images. Its results were compared to results from DECA extracting PNG files, operating in both linear and sampling mode, and specifying certain options for the file carving process in both modes, from the same disk images so as to measure how well the implemented PNG data detector works together with DECA. The results from experiments #1, #2, and #3 can be seen in Table 1, Table 2 and Table 3 below.

Table 1
The results from experiment #1.

File carver	PNG files extracted	Time taken (*s*)
Photorec	8	11
DECA (linear)	8	7
DECA (sampling)	2	1

Table 2
The results from experiment #2.

File carver	PNG files extracted	Time taken (*s*)
Photorec	2	3
DECA (linear)	5	28
DECA (sampling)	0	1

Table 3
The results from experiment #3.

File carver	PNG files extracted	Time taken (*s*)
Photorec	8761	380
DECA (linear)	8599	273
DECA (sampling)	5111	99
DECA (linear + specified partition)	8599	245
DECA (sampling + specified partition)	5111	97

In (Gladyshev & James, 2017), it was established that operating in sampling mode would extract less amount of files than most linear carvers, which was consistent with the results from these experiments. Also based on the results, knowing and specifying the likely partition or volume to find the desired data as part of the argument for DECA's PNG file carving operation will produce faster results. The results from the experiments show that DECA integrated with the PNG data detector works best in scenarios where PNG data are complete and are stored in contiguous data blocks.

References

Adler, M. et al., 1999. *Chunk Specifications.* [Online] Available at: http://www.libpng.org/pub/png/spec/1.2/PNG-Chunks.html[Accessed 10 May 2019].

Adler, M. et al., 2017. *Extensions to the PNG 1.2 Specification, Version 1.5.0.* [Online] Available at: https://ftp-osl.osuosl.org/pub/libpng/documents/pngext-1.5.0.html[Accessed 14 June 2019].

CGSecurity, 2019. *PhotoRec.* [Online] Available at: https://www.cgsecurity.org/wiki/PhotoRec[Accessed 17 June 2019].

Gladyshev, P. & James, J. I., 2017. Decision-theoretic file carving. *Digital Investigation,* Volume 22, pp. 41-61.

Gladyshev, P., James, J. I. & Odogwu, K. C., 2016. *Deca Source Code.* [Online] Available at: https://bitbucket.org/pavel_gladyshev/deca-dij.git.

National Institute of Standards and Technology, 2019. *Forensic Images Used for NIST/CFTT File Carving Test Reports.* [Online] Available at: https://www.cfreds.nist.gov/filecarvingtestreports.html[Accessed 16 June 2019].