Forensic String Search Tool Quirks

By

James R. Lyle

From the proceedings of

The Digital Forensic Research Conference

DFRWS 2019 USA

Portland, OR (July 15th - 19th)

FORENSIC STRING SEARCH TOOL QUIRKS

WHAT I LEARNED TESTING STRING SEARCH TOOLS

James R. Lyle

National Institute of Standards and Technology, 100 Bureau Drive Stop 8970, Gaithersburg, MD 20899-8970

Disclaimer

Certain trade names and company products are mentioned in the text or identified. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.

CFTT

The CFTT project at NIST develops methodologies for testing computer forensic tools. Currently there are CFTT methodologies for testing the following:

- Disk imaging*
- Write blocking*
- Deleted File Recovery
- File Carving
- Forensic Media Preparation
- Mobile Devices*
- String Searching

A variety of tools in each of these categories have been tested and observed flaws in the tools have been reported by the Department of Homeland Security (DHS) and the National Institute of Justice (NIJ). These results can be used as a basis for identifying the types of likely failures that occur in forensic tools.

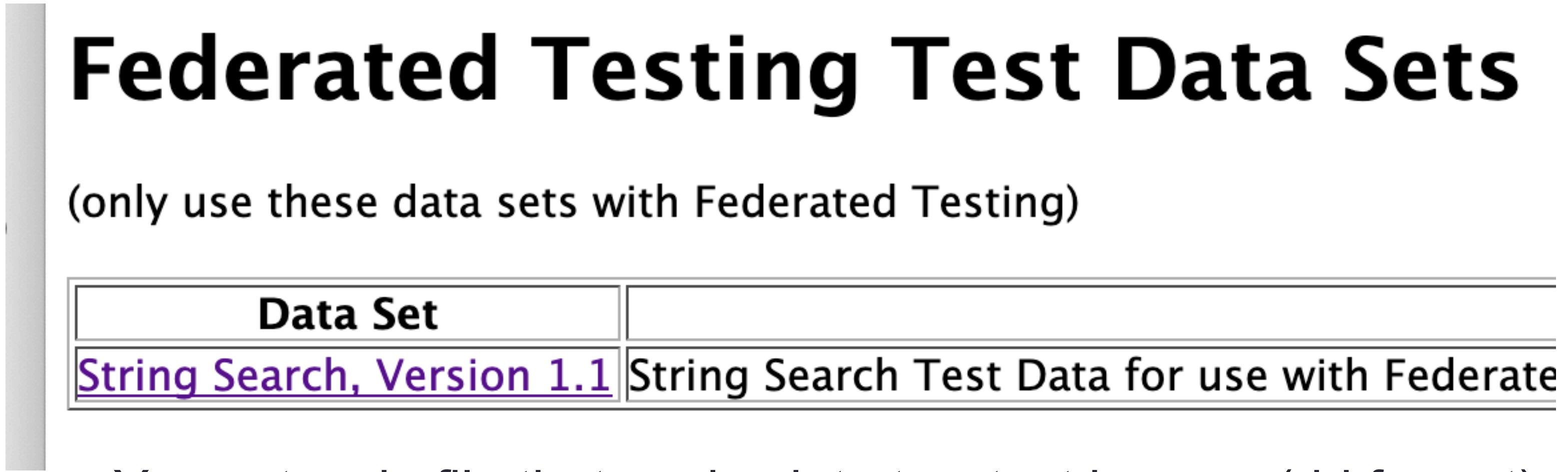
* Starred methods have been incorporated into Federated Testing

What String Search Features We Selected to Test

- Match case vs Ignore Case
- Match whole Words vs substrings
- Search engine/method: indexed vs live vs physical
- File systems: FAT32, ExFAT, NTFS, ext4, OSXJ, OSXC & APFS
- Encoding: ASCII, UTF-8, UTF-16 (BE & LE) with & without byte-order-mark
- Language: CJK, Latin with diacritics, non-Latin, right-to-left
- Live Files vs Deleted Files vs Unallocated Space
- Logical expressions
- Regular expressions
- Special Cases
  - Meta-data
  - Formatted documents (.doc, .docx, .html)
  - Small files in NTFS $MFT
  - Search target spans fragmentation
  - Stemming

Getting the NIST/CFTT String Search Data Set

- Download from www.cfreds.nist.gov
- Click on this link (below)



Federated Testing Test Data Sets

(only use these data sets with Federated Testing)

Data Set	
String Search, Version 1.1	String Search Test Data for use with Federate

- You get a zip file that unzips into two test images (dd format) and . . .
- One test image has MS Windows partitions (FAT, ExFAT & NTFS)
- The other image has Ext4, HFS+ & APFS formatted partitions
- Several files documenting test cases & expected results
- This is stand-alone and you don't need Federated Testing to run the tests

Tools Tested So Far . . .

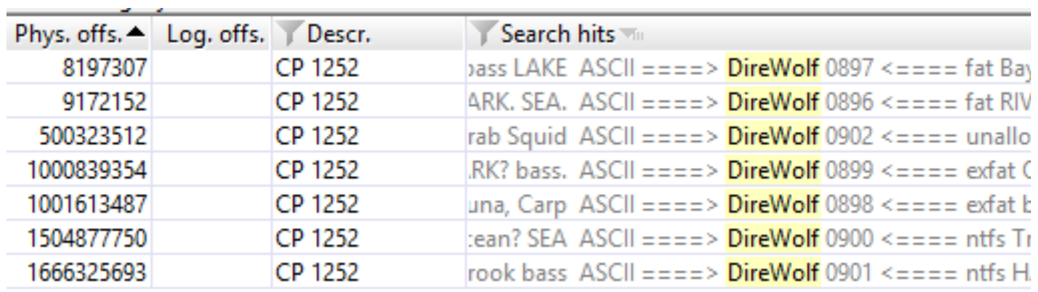
- Autopsy 4.6 (Test Report Posted November 2018)
- X-Ways 19.6 SR4 (Test Report Posted June 2019)
- FTK 7.0.0.163 (Test Report under review by vendor)
- BlackLight 2018R4 (Test Report under review by vendor)

Try a Search Tool – X-Ways

- Try to find a DireWolf (just in case “Winter is Coming”)
- Expected Results: 7 hits
  - Two hits in each partition, one active file, one deleted file
  - One hit in unallocated space

ID	String	Offset	File Name
0897	DireWolf	8,197,307	DELETED-Extinct-Lupus-fat-ascii.txt
0896	DireWolf	9,172,152	LIVE-Extinct-Lupus-fat-ascii.txt
0902	DireWolf	500,323,512	LIVE-Extinct-Lupus-unalloc-ascii.txt
0899	DireWolf	1,000,839,354	DELETED-Extinct-Lupus-exfat-ascii.txt
0898	DireWolf	1,001,613,487	LIVE-Extinct-Lupus-exfat-ascii.txt
0900	DireWolf	1,504,877,750	LIVE-Extinct-Lupus-ntfs-ascii.txt
0901	DireWolf	1,666,325,693	DELETED-Extinct-Lupus-ntfs-ascii.txt

Found 7 hits; this is what I Expected:
(X-Ways screen shot)



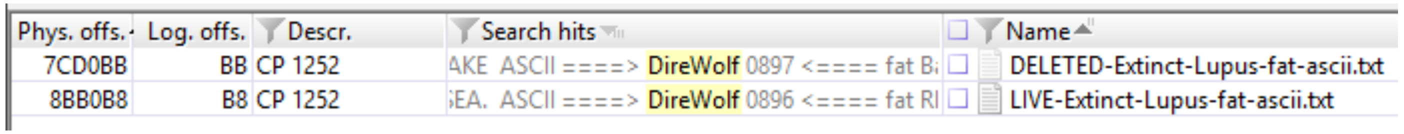
Phys. offs.	Log. offs.	Descr.	Search hits
8197307		CP 1252	ɔass LAKE ASCII ====> DireWolf 0897 <==== fat Bay
9172152		CP 1252	ARK. SEA. ASCII ====> DireWolf 0896 <==== fat RIV
500323512		CP 1252	rab Squid ASCII ====> DireWolf 0902 <==== unallo
1000839354		CP 1252	.RK? bass. ASCII ====> DireWolf 0899 <==== exfat (
1001613487		CP 1252	una, Carp ASCII ====> DireWolf 0898 <==== exfat b
1504877750		CP 1252	:ean? SEA ASCII ====> DireWolf 0900 <==== ntfs Tı
1666325693		CP 1252	rook bass ASCII ====> DireWolf 0901 <==== ntfs H

Wow, this is easy & simple. Are We Done?

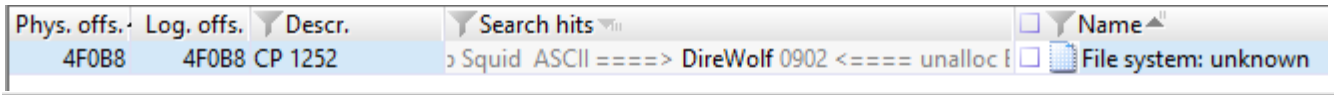
This was a physical search, let’s try another search engine . . .

More on X-Ways

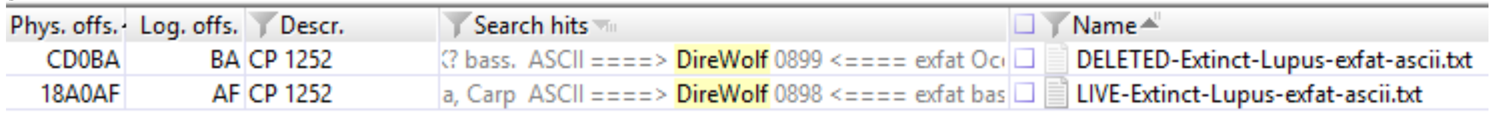
- That First X-Ways search was a physical search, one sector at a time.
- On X-Ways you can also search one file at a time: Logical Search



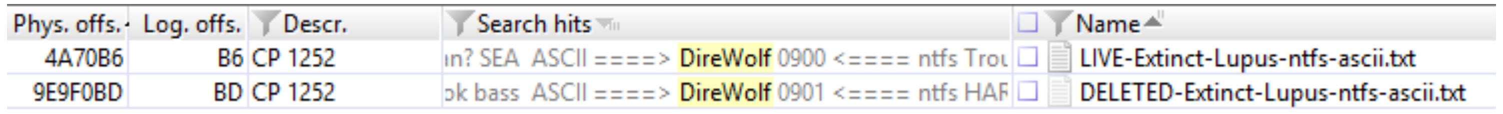
Phys. offs.	Log. offs.	Descr.	Search hits	☐	Name
7CD0BB	BB	CP 1252	AKE ASCII ====> DireWolf 0897 <==== fat B	☐	DELETED-Extinct-Lupus-fat-ascii.txt
8BB0B8	B8	CP 1252	;EA. ASCII ====> DireWolf 0896 <==== fat RI	☐	LIVE-Extinct-Lupus-fat-ascii.txt



Phys. offs.	Log. offs.	Descr.	Search hits	☐	Name
4F0B8	4F0B8	CP 1252	ɔ Squid ASCII ====> DireWolf 0902 <==== unalloc	☐	File system: unknown



Phys. offs.	Log. offs.	Descr.	Search hits	☐	Name
CD0BA	BA	CP 1252	;? bass. ASCII ====> DireWolf 0899 <==== exfat Oc	☐	DELETED-Extinct-Lupus-exfat-ascii.txt
18A0AF	AF	CP 1252	a, Carp ASCII ====> DireWolf 0898 <==== exfat bas	☐	LIVE-Extinct-Lupus-exfat-ascii.txt



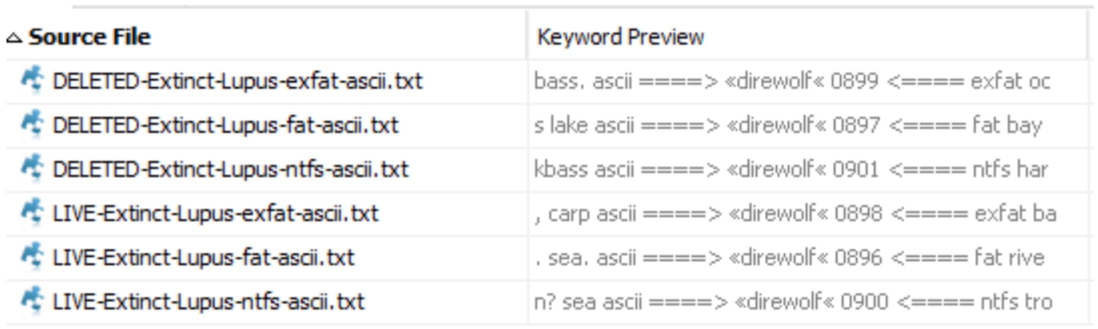
Phys. offs.	Log. offs.	Descr.	Search hits	☐	Name
4A70B6	B6	CP 1252	ın? SEA ASCII ====> DireWolf 0900 <==== ntfs Trou	☐	LIVE-Extinct-Lupus-ntfs-ascii.txt
9E9F0BD	BD	CP 1252	ɔk bass ASCII ====> DireWolf 0901 <==== ntfs HAF	☐	DELETED-Extinct-Lupus-ntfs-ascii.txt

- X-Ways can also do an indexed search
- Let's try another tool . . .

Let’s Try Another Tool – Autopsy 4.6

- Autopsy results:

First try . . .



Source File	Keyword Preview
DELETED-Extinct-Lupus-exfat-ascii.txt	bass. ascii ====> «direwolf« 0899 <==== exfat oc
DELETED-Extinct-Lupus-fat-ascii.txt	s lake ascii ====> «direwolf« 0897 <==== fat bay
DELETED-Extinct-Lupus-ntfs-ascii.txt	kbass ascii ====> «direwolf« 0901 <==== ntfs har
LIVE-Extinct-Lupus-exfat-ascii.txt	, carp ascii ====> «direwolf« 0898 <==== exfat ba
LIVE-Extinct-Lupus-fat-ascii.txt	. sea. ascii ====> «direwolf« 0896 <==== fat rive
LIVE-Extinct-Lupus-ntfs-ascii.txt	n? sea ascii ====> «direwolf« 0900 <==== ntfs tro

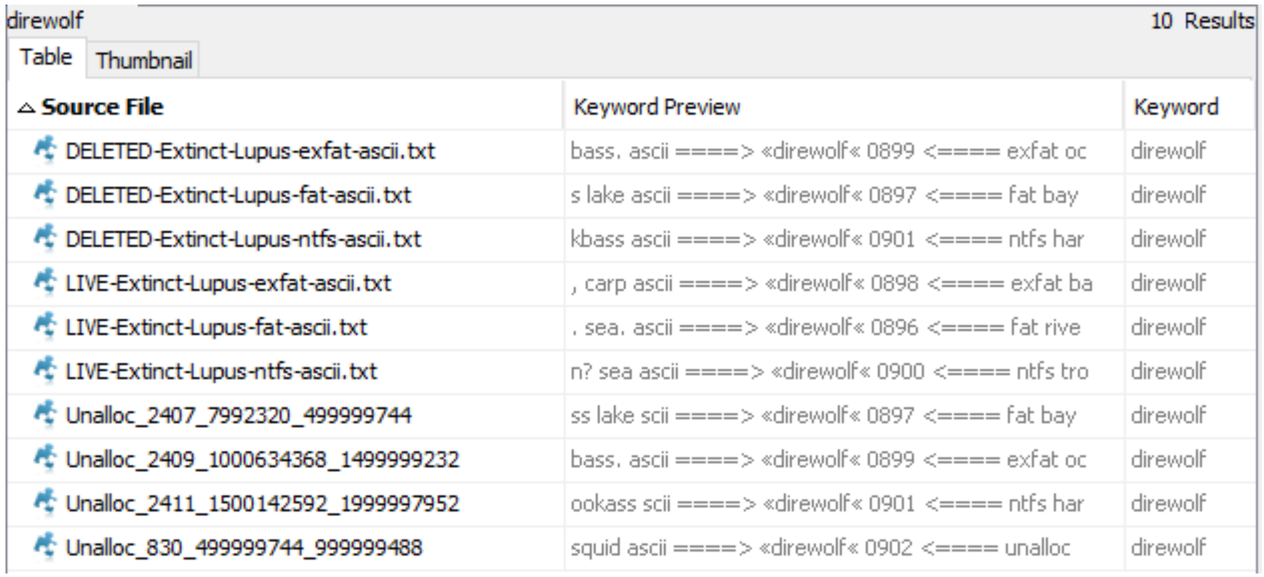
Oops, 6 hits, Did we miss one?

Try again . . . Now 10 hits, too many?

3 hits are reported twice!

Recovered deleted file is also unallocated space!!

How did I get the second result?



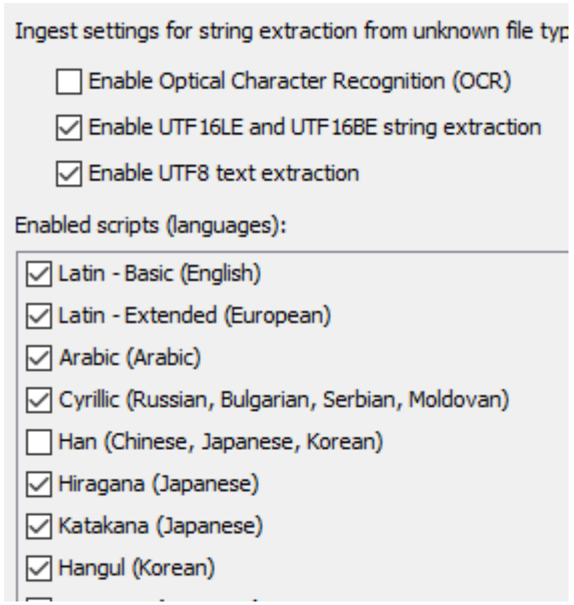
direwolf 10 Results

Table Thumbnail

Source File	Keyword Preview	Keyword
DELETED-Extinct-Lupus-exfat-ascii.txt	bass. ascii ====> «direwolf« 0899 <==== exfat oc	direwolf
DELETED-Extinct-Lupus-fat-ascii.txt	s lake ascii ====> «direwolf« 0897 <==== fat bay	direwolf
DELETED-Extinct-Lupus-ntfs-ascii.txt	kbass ascii ====> «direwolf« 0901 <==== ntfs har	direwolf
LIVE-Extinct-Lupus-exfat-ascii.txt	, carp ascii ====> «direwolf« 0898 <==== exfat ba	direwolf
LIVE-Extinct-Lupus-fat-ascii.txt	. sea. ascii ====> «direwolf« 0896 <==== fat rive	direwolf
LIVE-Extinct-Lupus-ntfs-ascii.txt	n? sea ascii ====> «direwolf« 0900 <==== ntfs tro	direwolf
Unalloc_2407_7992320_499999744	ss lake scii ====> «direwolf« 0897 <==== fat bay	direwolf
Unalloc_2409_1000634368_1499999232	bass. ascii ====> «direwolf« 0899 <==== exfat oc	direwolf
Unalloc_2411_1500142592_1999997952	ookass scii ====> «direwolf« 0901 <==== ntfs har	direwolf
Unalloc_830_499999744_999999488	squid ascii ====> «direwolf« 0902 <==== unalloc	direwolf

Autopsy 4.6 Search Settings

- Autopsy has selections for searching and indexing **Unallocated Space**
- If we select Han, ASCII string (and other stuff) is not found in unallocated space
- If we unselect UTF-8 & UTF-16, ASCII string not found in unallocated space
- To get the second result
  - uncheck Han and
  - check “enable” at least one of the “UTF” settings
- Later versions of Autopsy fixed this



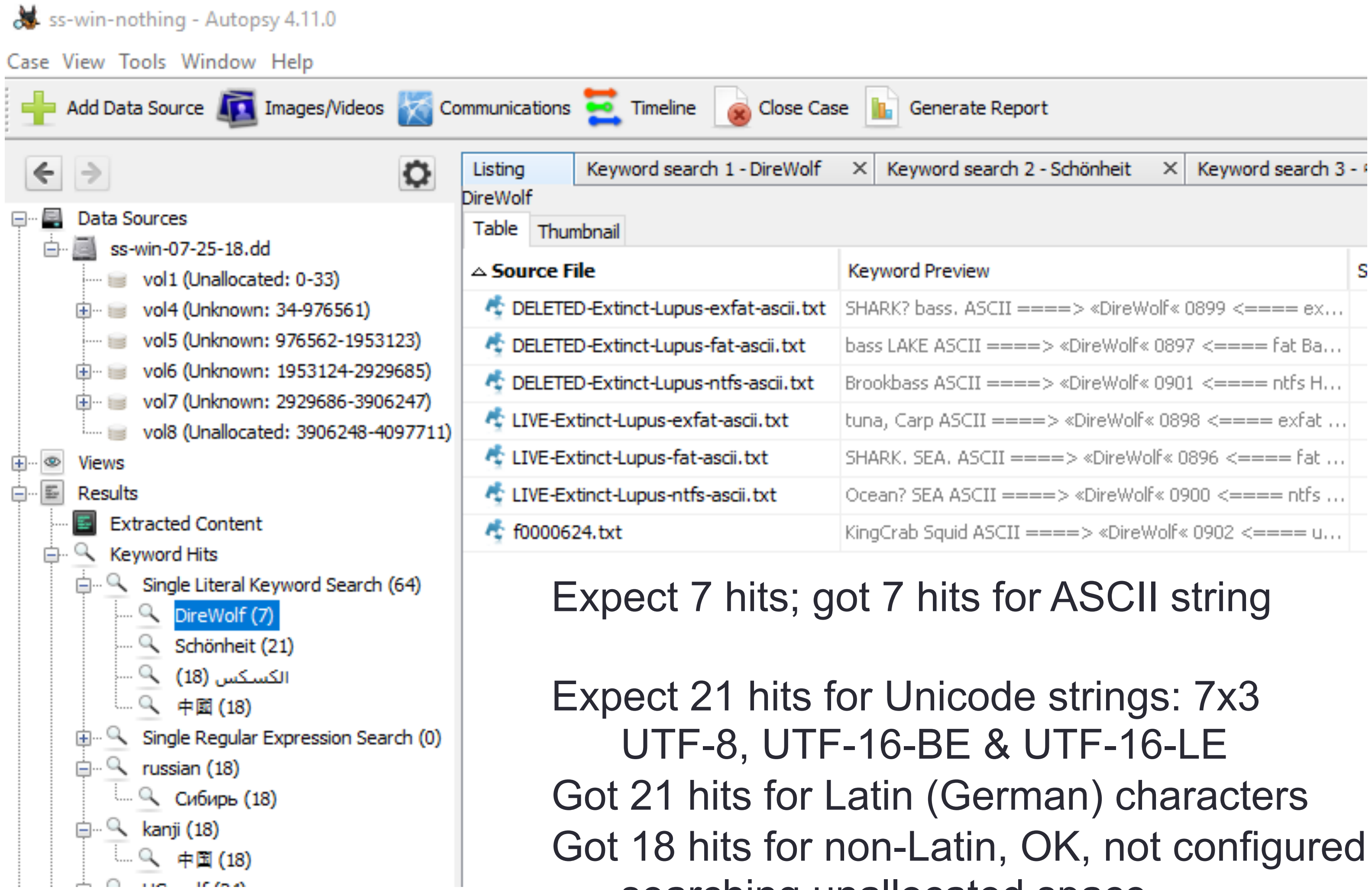
Ingest settings for string extraction from unknown file typ

- [ ] Enable Optical Character Recognition (OCR)
- [✓] Enable UTF16LE and UTF16BE string extraction
- [✓] Enable UTF8 text extraction

Enabled scripts (languages):

- [✓] Latin - Basic (English)
- [✓] Latin - Extended (European)
- [✓] Arabic (Arabic)
- [✓] Cyrillic (Russian, Bulgarian, Serbian, Moldovan)
- [ ] Han (Chinese, Japanese, Korean)
- [✓] Hiragana (Japanese)
- [✓] Katakana (Japanese)
- [✓] Hangul (Korean)

Autopsy 4.11 Results


ss-win-nothing - Autopsy 4.11.0

Case View Tools Window Help

Add Data Source | Images/Videos | Communications | Timeline | Close Case | Generate Report

- Data Sources
  - ss-win-07-25-18.dd
    - vol1 (Unallocated: 0-33)
    - vol4 (Unknown: 34-976561)
    - vol5 (Unknown: 976562-1953123)
    - vol6 (Unknown: 1953124-2929685)
    - vol7 (Unknown: 2929686-3906247)
    - vol8 (Unallocated: 3906248-4097711)
- Views
- Results
  - Extracted Content
  - Keyword Hits
    - Single Literal Keyword Search (64)
      - DireWolf (7)
      - Schönheit (21)
      - (18) الكسكس
      - 中国 (18)
    - Single Regular Expression Search (0)
    - russian (18)
      - Сибирь (18)
    - kanji (18)
      - 中国 (18)

Listing | Keyword search 1 - DireWolf | Keyword search 2 - Schönheit | Keyword search 3 -

DireWolf

Table | Thumbnail

Source File	Keyword Preview
DELETED-Extinct-Lupus-exfat-ascii.txt	SHARK? bass. ASCII ====> «DireWolf« 0899 <==== ex...
DELETED-Extinct-Lupus-fat-ascii.txt	bass LAKE ASCII ====> «DireWolf« 0897 <==== fat Ba...
DELETED-Extinct-Lupus-ntfs-ascii.txt	Brookbass ASCII ====> «DireWolf« 0901 <==== ntfs H...
LIVE-Extinct-Lupus-exfat-ascii.txt	tuna, Carp ASCII ====> «DireWolf« 0898 <==== exfat ...
LIVE-Extinct-Lupus-fat-ascii.txt	SHARK. SEA. ASCII ====> «DireWolf« 0896 <==== fat ...
LIVE-Extinct-Lupus-ntfs-ascii.txt	Ocean? SEA ASCII ====> «DireWolf« 0900 <==== ntfs ...
f0000624.txt	KingCrab Squid ASCII ====> «DireWolf« 0902 <==== u...

Expect 7 hits; got 7 hits for ASCII string

Expect 21 hits for Unicode strings: 7x3
UTF-8, UTF-16-BE & UTF-16-LE
Got 21 hits for Latin (German) characters
Got 18 hits for non-Latin, OK, not configured for searching unallocated space

General Observations about Quirks

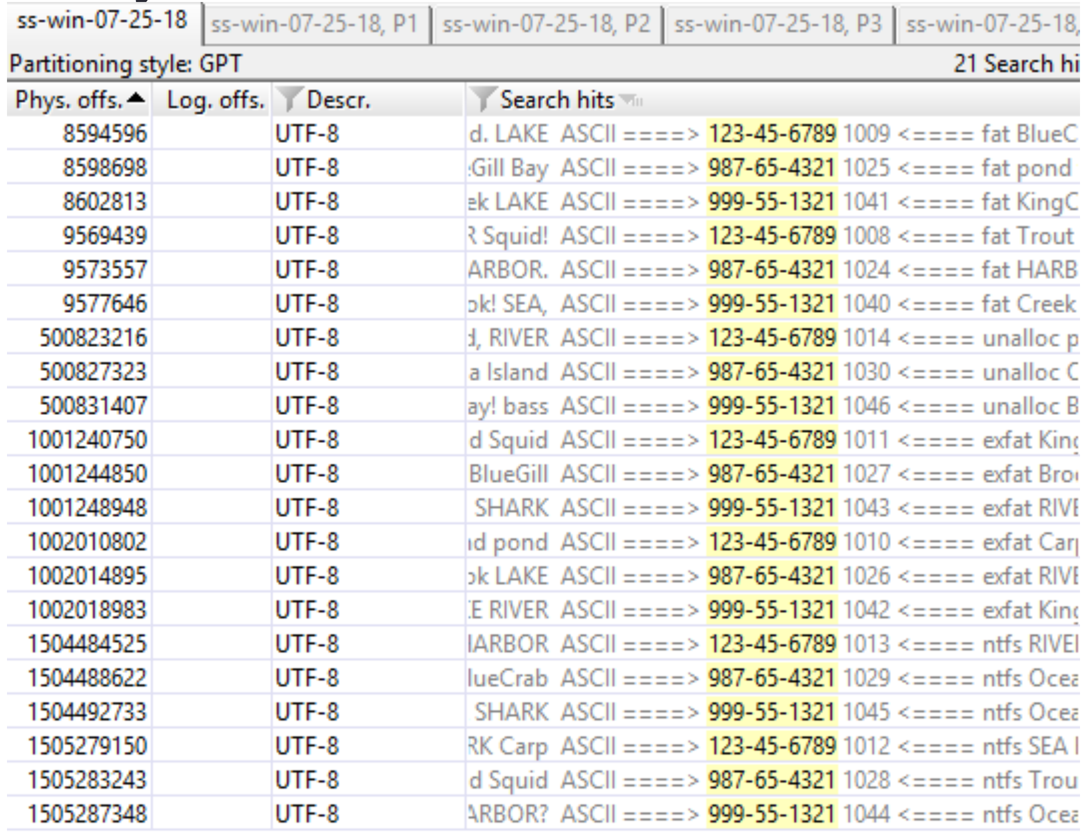
- Search Configuration has to be set with care
- Different search engines within a tool may give different results
- Meta-data quirks
- Mac (OSX) file system quirks
- Unsupported file systems (treated as unallocated space)
- Unicode Quirks

Some Other Observed Tool Behaviors

- Most tools could parse FAT, ExFAT, NTFS, ext4, journaled OSX and case-sensitive OSX partitions. Sometimes ExFAT or APFS not supported
- Usually found ASCII, UTF-8 & UTF-16, but sometimes failed for particular languages, to find UTF-16 strings
- Sometimes indexed search and live search have differences.
- Sometimes UTF-16BE reported as UTF-16LE and vice versa
- Usually 1-1 reporting of each hit to location, but sometimes reported as multiple hits
- One older tool version reported a corrupted name for some ExFAT files containing a hit
- One tool fails to render Korean UNICODE string correctly
- Some tools fail to ignore embedded HTML tags
- Most tools failed to recognize and decode docx file in unallocated space

Finding Social Security Numbers

- Tools often have built-in searches for interesting items like social security numbers, phone numbers, credit cards & IP addresses
- For example, Social Security search returns:
- For X-Ways . . .
- (Actually this was by a regular expression)
- 3 partitions x 3 strings 2 times per partition + 3 in unallocated = expect 21 hits



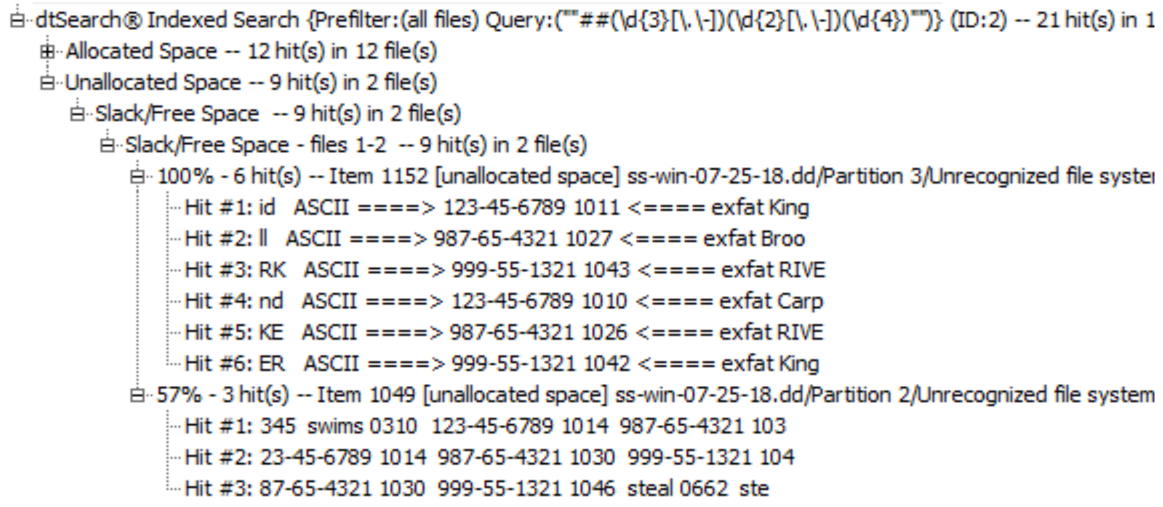
ss-win-07-25-18 | ss-win-07-25-18, P1 | ss-win-07-25-18, P2 | ss-win-07-25-18, P3 | ss-win-07-25-18,

Partitioning style: GPT 21 Search hi

Phys. offs.	Log. offs.	Descr.	Search hits
8594596		UTF-8	d. LAKE ASCII ====> 123-45-6789 1009 <==== fat BlueC
8598698		UTF-8	:Gill Bay ASCII ====> 987-65-4321 1025 <==== fat pond
8602813		UTF-8	ek LAKE ASCII ====> 999-55-1321 1041 <==== fat KingC
9569439		UTF-8	₹ Squid! ASCII ====> 123-45-6789 1008 <==== fat Trout
9573557		UTF-8	ARBOR. ASCII ====> 987-65-4321 1024 <==== fat HARB
9577646		UTF-8	ɔk! SEA, ASCII ====> 999-55-1321 1040 <==== fat Creek
500823216		UTF-8	ɟ, RIVER ASCII ====> 123-45-6789 1014 <==== unalloc p
500827323		UTF-8	a Island ASCII ====> 987-65-4321 1030 <==== unalloc C
500831407		UTF-8	ay! bass ASCII ====> 999-55-1321 1046 <==== unalloc B
1001240750		UTF-8	d Squid ASCII ====> 123-45-6789 1011 <==== exfat King
1001244850		UTF-8	BlueGill ASCII ====> 987-65-4321 1027 <==== exfat Bro
1001248948		UTF-8	SHARK ASCII ====> 999-55-1321 1043 <==== exfat RIVE
1002010802		UTF-8	ıd pond ASCII ====> 123-45-6789 1010 <==== exfat Car
1002014895		UTF-8	ɔk LAKE ASCII ====> 987-65-4321 1026 <==== exfat RIVE
1002018983		UTF-8	E RIVER ASCII ====> 999-55-1321 1042 <==== exfat King
1504484525		UTF-8	IARBOR ASCII ====> 123-45-6789 1013 <==== ntfs RIVE
1504488622		UTF-8	lueCrab ASCII ====> 987-65-4321 1029 <==== ntfs Ocea
1504492733		UTF-8	SHARK ASCII ====> 999-55-1321 1045 <==== ntfs Ocea
1505279150		UTF-8	RK Carp ASCII ====> 123-45-6789 1012 <==== ntfs SEA I
1505283243		UTF-8	d Squid ASCII ====> 987-65-4321 1028 <==== ntfs Trou
1505287348		UTF-8	ARBOR? ASCII ====> 999-55-1321 1044 <==== ntfs Ocea

Let’s try FTK -- Built-in Indexed Search

- FTK indexed search results:
- 12 hits in allocated space +
- 9 hits in unallocated space =
- Total of 21 hits



```
dtSearch® Indexed Search {Prefilter:(all files) Query:(""##(\d{3}[\.\-])(\d{2}[\.\-])(\d{4})"")} (ID:2) -- 21 hit(s) in 1
  Allocated Space -- 12 hit(s) in 12 file(s)
  Unallocated Space -- 9 hit(s) in 2 file(s)
    Slack/Free Space  -- 9 hit(s) in 2 file(s)
      Slack/Free Space - files 1-2  -- 9 hit(s) in 2 file(s)
        100% - 6 hit(s) -- Item 1152 [unallocated space] ss-win-07-25-18.dd/Partition 3/Unrecognized file syste
          Hit #1: id   ASCII ====> 123-45-6789 1011 <==== exfat King
          Hit #2: ll   ASCII ====> 987-65-4321 1027 <==== exfat Broo
          Hit #3: RK   ASCII ====> 999-55-1321 1043 <==== exfat RIVE
          Hit #4: nd   ASCII ====> 123-45-6789 1010 <==== exfat Carp
          Hit #5: KE   ASCII ====> 987-65-4321 1026 <==== exfat RIVE
          Hit #6: ER   ASCII ====> 999-55-1321 1042 <==== exfat King
        57% - 3 hit(s) -- Item 1049 [unallocated space] ss-win-07-25-18.dd/Partition 2/Unrecognized file system
          Hit #1: 345  swims 0310  123-45-6789 1014  987-65-4321 103
          Hit #2: 23-45-6789 1014  987-65-4321 1030  999-55-1321 104
          Hit #3: 87-65-4321 1030  999-55-1321 1046  steal 0662  ste
```

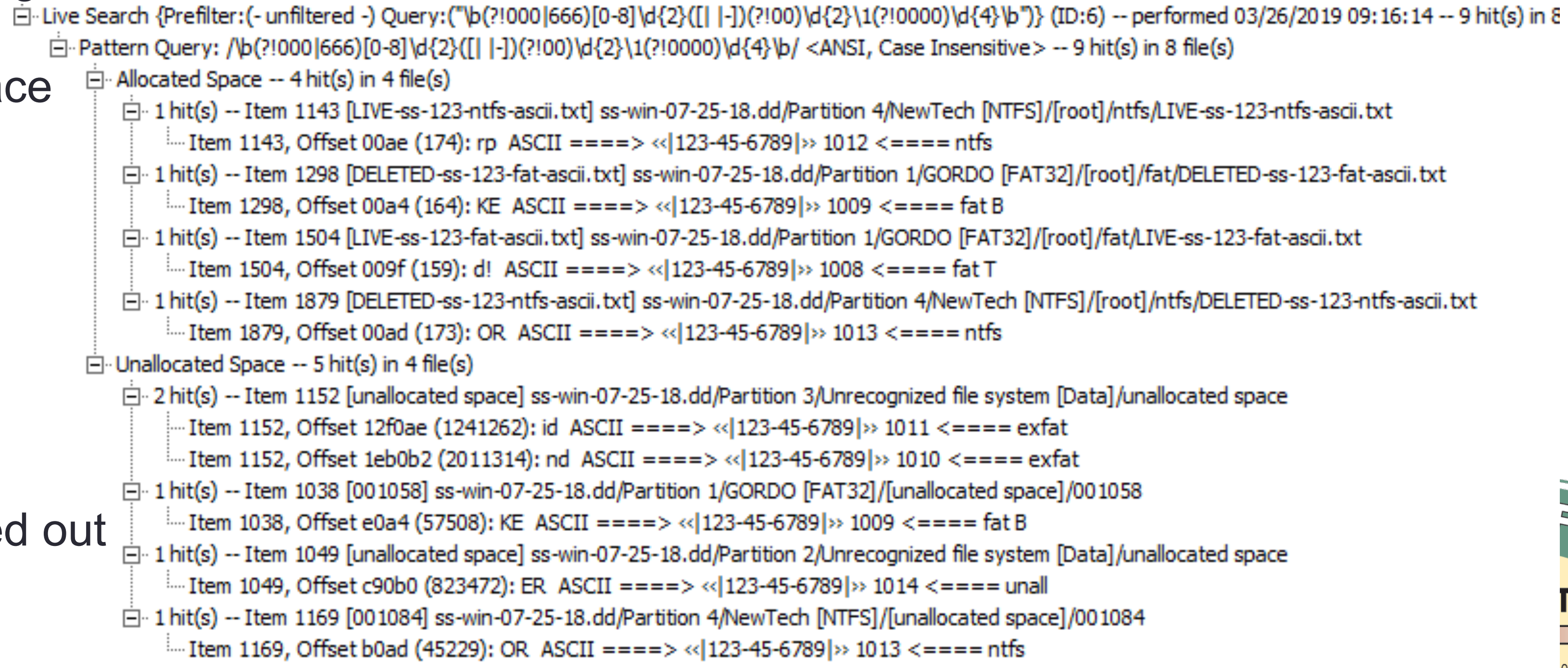
- Wait, wait. Shouldn’t it be :
  18 allocated + 3 unallocated?

- FTK does not support ExFAT (or APFS), so searched as unallocated space
- Also the presentation of the hits from partition 2 is a little unclear

More FTK Social Security – Built-in Live Search

Search results for FTK doing a LIVE search:

- 9 hits in allocated space
- 5 hits in unallocated space



```
⊟ Live Search {Prefilter:(- unfiltered -) Query:("\b(?!000|666)[0-8]\d{2}([| |-])(?!00)\d{2}\1(?!0000)\d{4}\b")} (ID:6) -- performed 03/26/2019 09:16:14 -- 9 hit(s) in 8
  ⊟ Pattern Query: /\b(?!000|666)[0-8]\d{2}([| |-])(?!00)\d{2}\1(?!0000)\d{4}\b/ <ANSI, Case Insensitive> -- 9 hit(s) in 8 file(s)
    ⊟ Allocated Space -- 4 hit(s) in 4 file(s)
      ⊟ 1 hit(s) -- Item 1143 [LIVE-ss-123-ntfs-ascii.txt] ss-win-07-25-18.dd/Partition 4/NewTech [NTFS]/[root]/ntfs/LIVE-ss-123-ntfs-ascii.txt
         Item 1143, Offset 00ae (174): rp  ASCII ====> «|123-45-6789|» 1012 <==== ntfs
      ⊟ 1 hit(s) -- Item 1298 [DELETED-ss-123-fat-ascii.txt] ss-win-07-25-18.dd/Partition 1/GORDO [FAT32]/[root]/fat/DELETED-ss-123-fat-ascii.txt
         Item 1298, Offset 00a4 (164): KE  ASCII ====> «|123-45-6789|» 1009 <==== fat B
      ⊟ 1 hit(s) -- Item 1504 [LIVE-ss-123-fat-ascii.txt] ss-win-07-25-18.dd/Partition 1/GORDO [FAT32]/[root]/fat/LIVE-ss-123-fat-ascii.txt
         Item 1504, Offset 009f (159): d!  ASCII ====> «|123-45-6789|» 1008 <==== fat T
      ⊟ 1 hit(s) -- Item 1879 [DELETED-ss-123-ntfs-ascii.txt] ss-win-07-25-18.dd/Partition 4/NewTech [NTFS]/[root]/ntfs/DELETED-ss-123-ntfs-ascii.txt
         Item 1879, Offset 00ad (173): OR  ASCII ====> «|123-45-6789|» 1013 <==== ntfs
    ⊟ Unallocated Space -- 5 hit(s) in 4 file(s)
      ⊟ 2 hit(s) -- Item 1152 [unallocated space] ss-win-07-25-18.dd/Partition 3/Unrecognized file system [Data]/unallocated space
         Item 1152, Offset 12f0ae (1241262): id  ASCII ====> «|123-45-6789|» 1011 <==== exfat
         Item 1152, Offset 1eb0b2 (2011314): nd  ASCII ====> «|123-45-6789|» 1010 <==== exfat
      ⊟ 1 hit(s) -- Item 1038 [001058] ss-win-07-25-18.dd/Partition 1/GORDO [FAT32]/[unallocated space]/001058
         Item 1038, Offset e0a4 (57508): KE  ASCII ====> «|123-45-6789|» 1009 <==== fat B
      ⊟ 1 hit(s) -- Item 1049 [unallocated space] ss-win-07-25-18.dd/Partition 2/Unrecognized file system [Data]/unallocated space
         Item 1049, Offset c90b0 (823472): ER  ASCII ====> «|123-45-6789|» 1014 <==== unall
      ⊟ 1 hit(s) -- Item 1169 [001084] ss-win-07-25-18.dd/Partition 4/NewTech [NTFS]/[unallocated space]/001084
         Item 1169, Offset b0ad (45229): OR  ASCII ====> «|123-45-6789|» 1013 <==== ntfs
```

- Where did the other two target strings go?
- 987-65-4321 & 999-55-1321

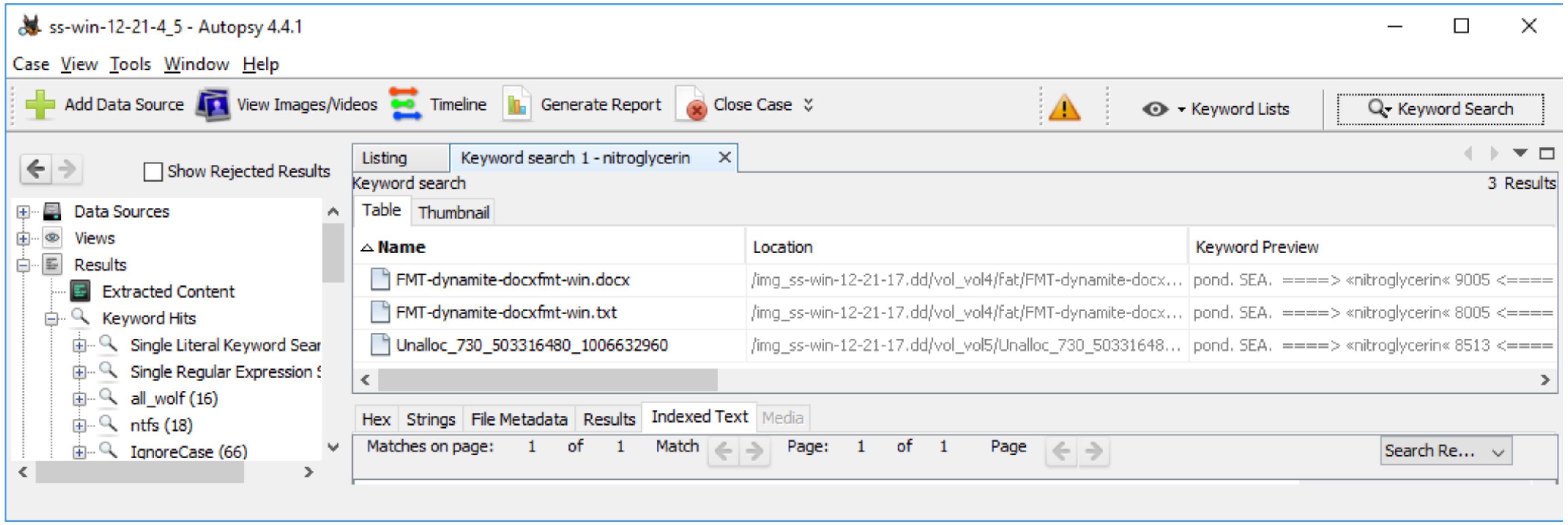
- Not valid (9xx), so filtered out
- But wait, if no SSN, IRS assigns ITIN

- An ITIN is a 9-digit number, beginning with the number "9", formatted like an SSN (NNN-NN-NNNN).

Searching Formatted Text – MS Word, HTML

- Each string appears four times
  - Plain Text in FAT partition
  - Formatted Text in FAT partition
  - Plain Text in unallocated space
  - Formatted Text in unallocated space
- Formatting schemes used
  - MS Word .doc in UTF-8 and .doc in UTF-16 & .docx
  - HTML
- Part of the string is formatted bold and underlined
  - **Cross**Bow HTML <u><b>Cross</b></u>Bow
  - **Nitro**glycerin DOCX
  - **Shot**gun DOC

Formatted Text Searches – Find **nitro**glycerin


ss-win-12-21-4_5 - Autopsy 4.4.1

Case View Tools Window Help

Add Data Source | View Images/Videos | Timeline | Generate Report | Close Case | Keyword Lists | Keyword Search

Show Rejected Results

- Data Sources
- Views
- Results
  - Extracted Content
  - Keyword Hits
    - Single Literal Keyword Sear
    - Single Regular Expression S
    - all_wolf (16)
    - ntfs (18)
    - IgnoreCase (66)

Listing | Keyword search 1 - nitroglycerin

Keyword search 3 Results

Table | Thumbnail

Name	Location	Keyword Preview
FMT-dynamite-docxfmt-win.docx	/img_ss-win-12-21-17.dd/vol_vol4/fat/FMT-dynamite-docx...	pond. SEA. ====> «nitroglycerin« 9005 <====
FMT-dynamite-docxfmt-win.txt	/img_ss-win-12-21-17.dd/vol_vol4/fat/FMT-dynamite-docx...	pond. SEA. ====> «nitroglycerin« 8005 <====
Unalloc_730_503316480_1006632960	/img_ss-win-12-21-17.dd/vol_vol5/Unalloc_730_50331648...	pond. SEA. ====> «nitroglycerin« 8513 <====

Hex | Strings | File Metadata | Results | Indexed Text | Media

Matches on page: 1 of 1 Match Page: 1 of 1 Page Search Re...

The string nitroglycerin appears 4 times, note **nitro** has embedded tags:

- Text in the FAT Partition (8005) and in unallocated space (8513)
- Formatted text in a docx file: **nitro**glycerin (9005 in FAT and 9513 in unallocated space.
- This tool found formatted text in FAT, but no tool found string in unallocated space.
- The docx file in unallocated space needs to be carved and then searched.

Unicode Quirks

- We tested for strings in the most common representation:
  - in UTF-16-BE,
  - UTF-16-LE &
  - UTF-8 (Overlaps with ASCII)
- We tested Unicode features:
  - Byte-order-mark (UTF-16)
  - Normalization (Combining characters & ligatures)
- We did not test other representations, e.g.,
  - UTF-7 or UTF-32
  - EBCDIC
  - ISO 8859-2 through ISO 8859-16
  - Shift-JIS (Japan)
  - Guobaio (China)
  - Big5 (Taiwan)

Unicode Background

- Determining if Unicode UTF-16 text is UTF-16-BE or UTF-16-LE is problematic for some text samples, especially for Latin based characters, because a one-byte shift in starting point for a string can align with either representation. For example, consider the hex representing the string "Schönheit" in UTF-16:
-
- **00 53 00 63 00 68 00 f6 00 6e 00 68 00 65 00 69 00 74 00**
- **S c h o: n h e i t**
-
- If you start the match with 00 53 00 63 00 . . . then it is UTF-16-BE, but
- If you start the match with 53 00 63 00 . . . then it is UTF-16-LE, so without any other information is could be either BE or LE. This is an artifact of UTF-16 characters that have a first byte of zero for the big-endian representation (as in Latin based characters).

Where is Buzz Lightyear?

- Buzz is trying to get to “infinity” (and maybe beyond). . .
- Expected Results: 49 unique strings.
- Let’s try Autopsy 4.10 . . .
- Tool reports 46 strings, but . . .
- 4 of the hits are in unalloc space
- 3 of these hits are duplicates of hits in deleted files (46 – 3 => 43)
- The other unallocated hits should have 7 hits (43 + 7 – 1 => 49)

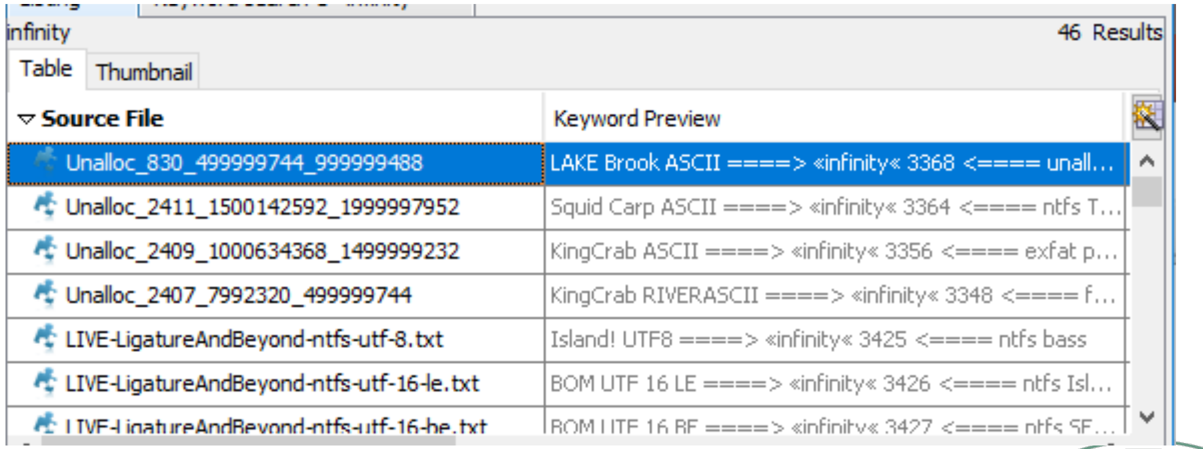

infinity 46 Results

Table Thumbnail

Source File	Keyword Preview
Unalloc_830_499999744_999999488	LAKE Brook ASCII ====> «infinity« 3368 <==== unall…
Unalloc_2411_1500142592_1999997952	Squid Carp ASCII ====> «infinity« 3364 <==== ntfs T…
Unalloc_2409_1000634368_1499999232	KingCrab ASCII ====> «infinity« 3356 <==== exfat p…
Unalloc_2407_7992320_499999744	KingCrab RIVERASCII ====> «infinity« 3348 <==== f…
LIVE-LigatureAndBeyond-ntfs-utf-8.txt	Island! UTF8 ====> «infinity« 3425 <==== ntfs bass
LIVE-LigatureAndBeyond-ntfs-utf-16-le.txt	BOM UTF 16 LE ====> «infinity« 3426 <==== ntfs Isl…
LIVE-LigatureAndBeyond-ntfs-utf-16-be.txt	BOM UTF 16 BE ====> «infinity« 3427 <==== ntfs SF…

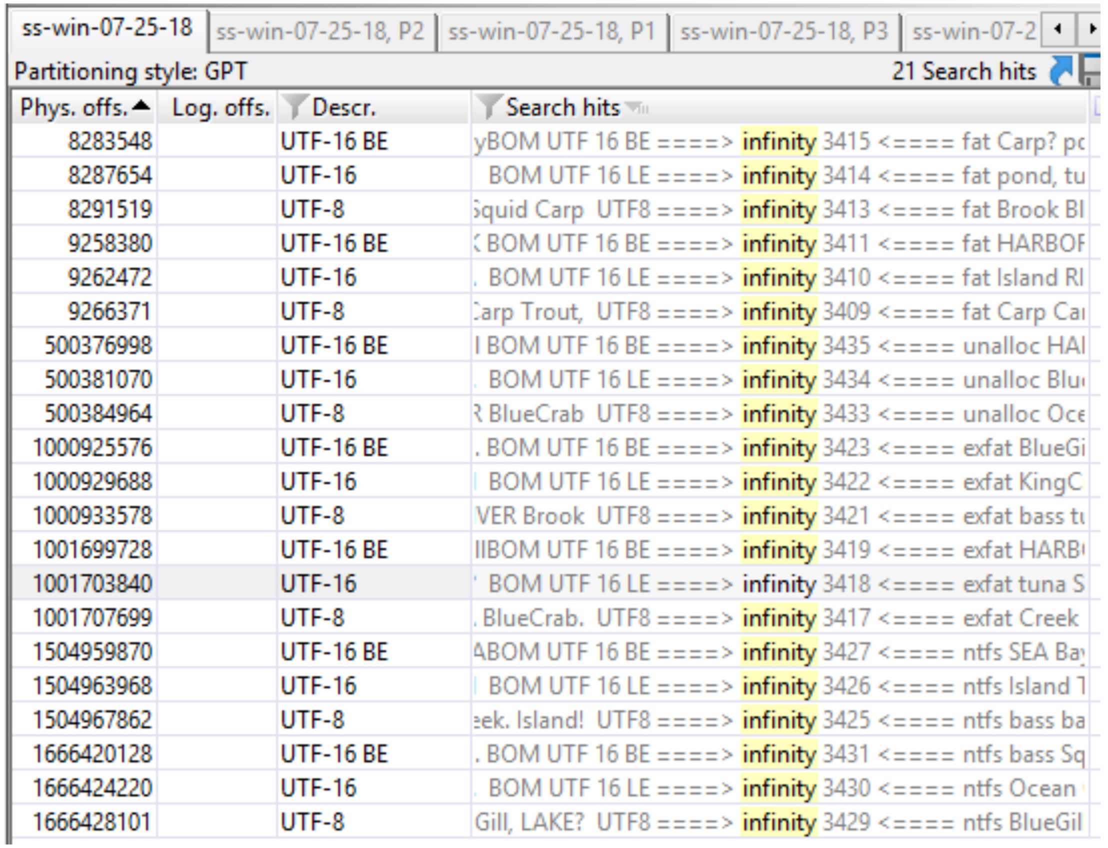
What’s a Ligature?

- Compare:
  - I n f i n i t y
  - I n ﬁ n i t y
- English has several ligatures: ﬀ, ﬂ, ﬃ, ﬄ, Æ, æ, Œ, . . ., etc
- A single byte code may represent more than one letter

- Guess what happens in German, French or Spanish . . . .
- Umlaut (Schönheit) , accents, tilde (cañón) . . .

More Buzz

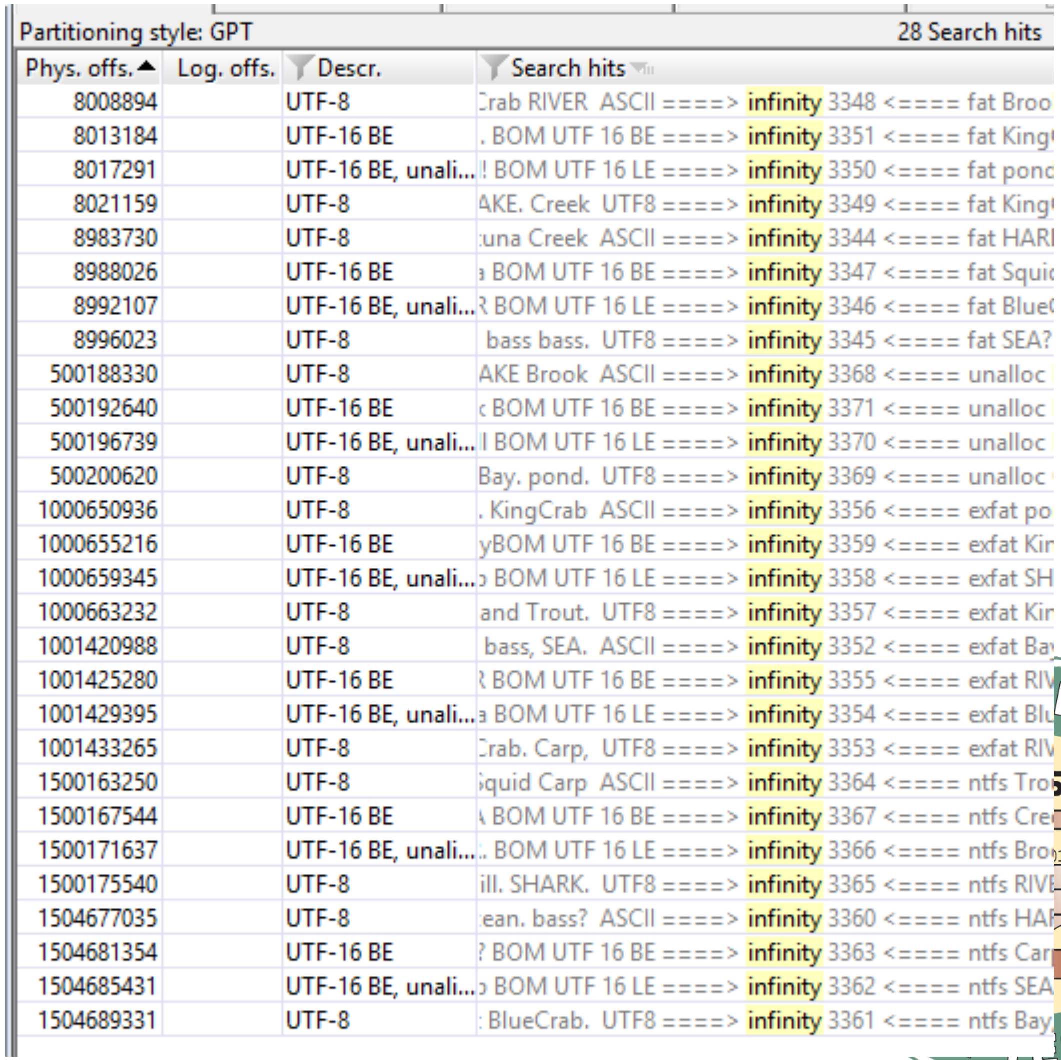
X-Ways results:
Of the 49 expected hits, 21 hits are with ligature and 28 Hits are without a ligature.



ss-win-07-25-18 | ss-win-07-25-18, P2 | ss-win-07-25-18, P1 | ss-win-07-25-18, P3 | ss-win-07-2

Partitioning style: GPT — 21 Search hits

Phys. offs.	Log. offs.	Descr.	Search hits
8283548		UTF-16 BE	yBOM UTF 16 BE ====> infinity 3415 <==== fat Carp? pc
8287654		UTF-16	BOM UTF 16 LE ====> infinity 3414 <==== fat pond, tu
8291519		UTF-8	Squid Carp UTF8 ====> infinity 3413 <==== fat Brook Bl
9258380		UTF-16 BE	K BOM UTF 16 BE ====> infinity 3411 <==== fat HARBOF
9262472		UTF-16	. BOM UTF 16 LE ====> infinity 3410 <==== fat Island RI
9266371		UTF-8	Carp Trout, UTF8 ====> infinity 3409 <==== fat Carp Cai
500376998		UTF-16 BE	I BOM UTF 16 BE ====> infinity 3435 <==== unalloc HAI
500381070		UTF-16	. BOM UTF 16 LE ====> infinity 3434 <==== unalloc Blu
500384964		UTF-8	R BlueCrab UTF8 ====> infinity 3433 <==== unalloc Oce
1000925576		UTF-16 BE	. BOM UTF 16 BE ====> infinity 3423 <==== exfat BlueGi
1000929688		UTF-16	BOM UTF 16 LE ====> infinity 3422 <==== exfat KingC
1000933578		UTF-8	VER Brook UTF8 ====> infinity 3421 <==== exfat bass tu
1001699728		UTF-16 BE	IIBOM UTF 16 BE ====> infinity 3419 <==== exfat HARB
1001703840		UTF-16	' BOM UTF 16 LE ====> infinity 3418 <==== exfat tuna S
1001707699		UTF-8	. BlueCrab. UTF8 ====> infinity 3417 <==== exfat Creek
1504959870		UTF-16 BE	ABOM UTF 16 BE ====> infinity 3427 <==== ntfs SEA Bay
1504963968		UTF-16	BOM UTF 16 LE ====> infinity 3426 <==== ntfs Island T
1504967862		UTF-8	eek. Island! UTF8 ====> infinity 3425 <==== ntfs bass ba
1666420128		UTF-16 BE	. BOM UTF 16 BE ====> infinity 3431 <==== ntfs bass Sq
1666424220		UTF-16	. BOM UTF 16 LE ====> infinity 3430 <==== ntfs Ocean
1666428101		UTF-8	Gill, LAKE? UTF8 ====> infinity 3429 <==== ntfs BlueGil



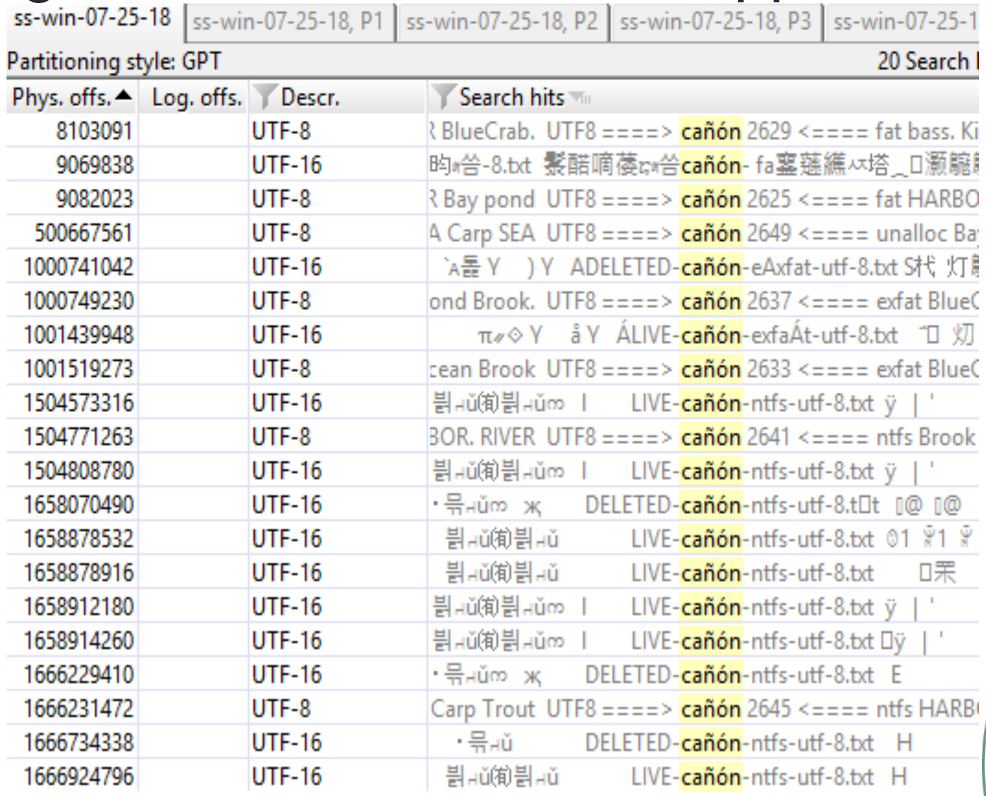
Partitioning style: GPT — 28 Search hits

Phys. offs.	Log. offs.	Descr.	Search hits
8008894		UTF-8	Crab RIVER ASCII ====> infinity 3348 <==== fat Broo
8013184		UTF-16 BE	. BOM UTF 16 BE ====> infinity 3351 <==== fat King
8017291		UTF-16 BE, unali...	! BOM UTF 16 LE ====> infinity 3350 <==== fat ponc
8021159		UTF-8	AKE. Creek UTF8 ====> infinity 3349 <==== fat King
8983730		UTF-8	una Creek ASCII ====> infinity 3344 <==== fat HARI
8988026		UTF-16 BE	a BOM UTF 16 BE ====> infinity 3347 <==== fat Squic
8992107		UTF-16 BE, unali...	R BOM UTF 16 LE ====> infinity 3346 <==== fat Blue
8996023		UTF-8	bass bass. UTF8 ====> infinity 3345 <==== fat SEA?
500188330		UTF-8	AKE Brook ASCII ====> infinity 3368 <==== unalloc
500192640		UTF-16 BE	c BOM UTF 16 BE ====> infinity 3371 <==== unalloc
500196739		UTF-16 BE, unali...	l BOM UTF 16 LE ====> infinity 3370 <==== unalloc
500200620		UTF-8	Bay. pond. UTF8 ====> infinity 3369 <==== unalloc
1000650936		UTF-8	. KingCrab ASCII ====> infinity 3356 <==== exfat po
1000655216		UTF-16 BE	yBOM UTF 16 BE ====> infinity 3359 <==== exfat Kir
1000659345		UTF-16 BE, unali...	o BOM UTF 16 LE ====> infinity 3358 <==== exfat SH
1000663232		UTF-8	and Trout. UTF8 ====> infinity 3357 <==== exfat Kir
1001420988		UTF-8	bass, SEA. ASCII ====> infinity 3352 <==== exfat Bay
1001425280		UTF-16 BE	R BOM UTF 16 BE ====> infinity 3355 <==== exfat RIV
1001429395		UTF-16 BE, unali...	a BOM UTF 16 LE ====> infinity 3354 <==== exfat Blu
1001433265		UTF-8	Crab. Carp, UTF8 ====> infinity 3353 <==== exfat RIV
1500163250		UTF-8	Squid Carp ASCII ====> infinity 3364 <==== ntfs Tro
1500167544		UTF-16 BE	A BOM UTF 16 BE ====> infinity 3367 <==== ntfs Cre
1500171637		UTF-16 BE, unali...	. BOM UTF 16 LE ====> infinity 3366 <==== ntfs Bro
1500175540		UTF-8	ill. SHARK. UTF8 ====> infinity 3365 <==== ntfs RIVE
1504677035		UTF-8	ean. bass? ASCII ====> infinity 3360 <==== ntfs HAF
1504681354		UTF-16 BE	? BOM UTF 16 BE ====> infinity 3363 <==== ntfs Car
1504685431		UTF-16 BE, unali...	o BOM UTF 16 LE ====> infinity 3362 <==== ntfs SEA
1504689331		UTF-8	: BlueCrab. UTF8 ====> infinity 3361 <==== ntfs Bay

Meta-Data on Windows (FAT, ExFAT & NTFS)

- A target string might be a substring of a file name. What happens then?
- Let's try "cañón" (Expect 7 hits + some meta-data hits)
- We got the 7 and then some meta-data

File System	Meta Data Count
FAT	1
ExFAT	2
NTFS	10


ss-win-07-25-18 | ss-win-07-25-18, P1 | ss-win-07-25-18, P2 | ss-win-07-25-18, P3 | ss-win-07-25-1

Partitioning style: GPT 20 Search

Phys. offs.	Log. offs.	Descr.	Search hits
8103091		UTF-8	R BlueCrab. UTF8 ====> cañón 2629 <==== fat bass. Ki
9069838		UTF-16	昀㈀-8.txt 鬏酤嘀蔆㈀cañón- fa塞莚纎ㅅ塔_口灦鶵
9082023		UTF-8	R Bay pond UTF8 ====> cañón 2625 <==== fat HARBO
500667561		UTF-8	A Carp SEA UTF8 ====> cañón 2649 <==== unalloc Ba
1000741042		UTF-16	`A壨 Y) Y ADELETED-cañón-eAxfat-utf-8.txt S朩 灯
1000749230		UTF-8	ond Brook. UTF8 ====> cañón 2637 <==== exfat Blue(
1001439948		UTF-16	π⫽◇ Y å Y ÁLIVE-cañón-exfaÁt-utf-8.txt 'ロ 炈
1001519273		UTF-8	cean Brook UTF8 ====> cañón 2633 <==== exfat Blue(
1504573316		UTF-16	뷝˨ŭ㈲뷝˨ŭ∞ I LIVE-cañón-ntfs-utf-8.txt ÿ | '
1504771263		UTF-8	BOR. RIVER UTF8 ====> cañón 2641 <==== ntfs Brook
1504808780		UTF-16	뷝˨ŭ㈲뷝˨ŭ∞ I LIVE-cañón-ntfs-utf-8.txt ÿ | '
1658070490		UTF-16	·뮤˨ŭ∞ ж DELETED-cañón-ntfs-utf-8.t□t ▯@ ▯@
1658878532		UTF-16	뷝˨ŭ㈲뷝˨ŭ LIVE-cañón-ntfs-utf-8.txt ▯1 ▯1 ▯
1658878916		UTF-16	뷝˨ŭ㈲뷝˨ŭ LIVE-cañón-ntfs-utf-8.txt □渠
1658912180		UTF-16	뷝˨ŭ㈲뷝˨ŭ∞ I LIVE-cañón-ntfs-utf-8.txt ÿ | '
1658914260		UTF-16	뷝˨ŭ㈲뷝˨ŭ∞ I LIVE-cañón-ntfs-utf-8.txt □ÿ | '
1666229410		UTF-16	·뮤˨ŭ∞ ж DELETED-cañón-ntfs-utf-8.txt E
1666231472		UTF-8	Carp Trout UTF8 ====> cañón 2645 <==== ntfs HARB
1666734338		UTF-16	·뮤˨ŭ DELETED-cañón-ntfs-utf-8.txt H
1666924796		UTF-16	뷝˨ŭ㈲뷝˨ŭ LIVE-cañón-ntfs-utf-8.txt H

- BlackLight found 2 meta-data instances in the FAT file system

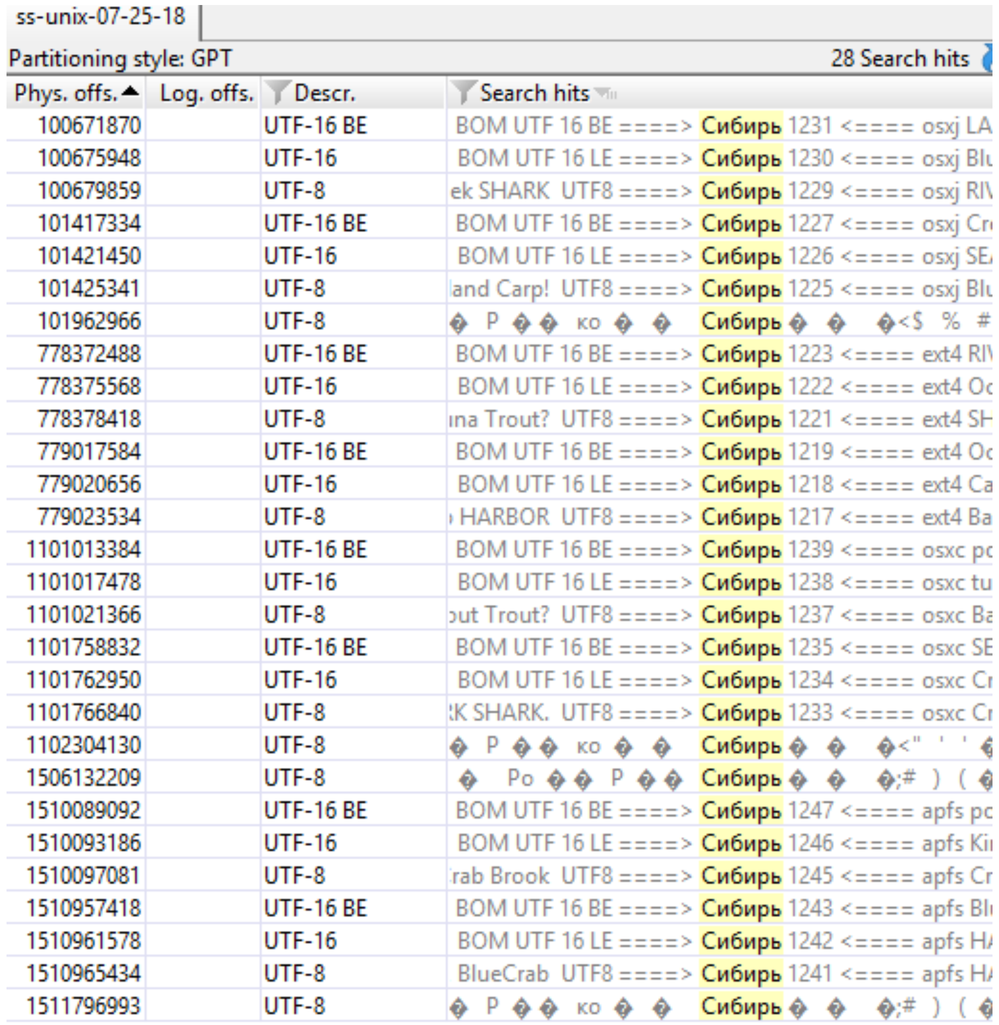
A Mystery

- On careful examination of locations of target strings on mac file systems, some strings have an extra instance in the image.
- Usually this is a word with non-Latin characters that seems to be in some sort of index or data-base. (spotlight?)
- Most forensic tools find the extra instance, some don’t ever find the string, other tools with multiple search engines find the string with one engine, but not the other.

What is this in the Unix-like file Systems?

- Should have 24 hits 3x2x4, but X-Ways reports 28
- The extra 4 all come from Mac
  - OSXJ – 1
  - OSXC – 1
  - APFS – 2

- OSX**J** is journaled, case insensitive
- OSX**C** is journaled, case sensitive
- APFS is (new) Apple File System



ss-unix-07-25-18

Partitioning style: GPT 28 Search hits

Phys. offs.	Log. offs.	Descr.	Search hits
100671870		UTF-16 BE	BOM UTF 16 BE ====> **Сибирь** 1231 <==== osxj LA
100675948		UTF-16	BOM UTF 16 LE ====> **Сибирь** 1230 <==== osxj Blu
100679859		UTF-8	ek SHARK UTF8 ====> **Сибирь** 1229 <==== osxj RI\
101417334		UTF-16 BE	BOM UTF 16 BE ====> **Сибирь** 1227 <==== osxj Cr
101421450		UTF-16	BOM UTF 16 LE ====> **Сибирь** 1226 <==== osxj SE
101425341		UTF-8	land Carp! UTF8 ====> **Сибирь** 1225 <==== osxj Blu
101962966		UTF-8	� P � � ко � � **Сибирь** � � �<$ % #
778372488		UTF-16 BE	BOM UTF 16 BE ====> **Сибирь** 1223 <==== ext4 RI\
778375568		UTF-16	BOM UTF 16 LE ====> **Сибирь** 1222 <==== ext4 Oc
778378418		UTF-8	ına Trout? UTF8 ====> **Сибирь** 1221 <==== ext4 SH
779017584		UTF-16 BE	BOM UTF 16 BE ====> **Сибирь** 1219 <==== ext4 Oc
779020656		UTF-16	BOM UTF 16 LE ====> **Сибирь** 1218 <==== ext4 Ca
779023534		UTF-8	HARBOR UTF8 ====> **Сибирь** 1217 <==== ext4 Ba
1101013384		UTF-16 BE	BOM UTF 16 BE ====> **Сибирь** 1239 <==== osxc pc
1101017478		UTF-16	BOM UTF 16 LE ====> **Сибирь** 1238 <==== osxc tu
1101021366		UTF-8	out Trout? UTF8 ====> **Сибирь** 1237 <==== osxc Ba
1101758832		UTF-16 BE	BOM UTF 16 BE ====> **Сибирь** 1235 <==== osxc SE
1101762950		UTF-16	BOM UTF 16 LE ====> **Сибирь** 1234 <==== osxc Cr
1101766840		UTF-8	:K SHARK. UTF8 ====> **Сибирь** 1233 <==== osxc Cr
1102304130		UTF-8	� P � � ко � � **Сибирь** � � �<" ' ' �
1506132209		UTF-8	� Po � � P � � **Сибирь** � � �;#) (�
1510089092		UTF-16 BE	BOM UTF 16 BE ====> **Сибирь** 1247 <==== apfs pc
1510093186		UTF-16	BOM UTF 16 LE ====> **Сибирь** 1246 <==== apfs Kir
1510097081		UTF-8	rab Brook UTF8 ====> **Сибирь** 1245 <==== apfs Cr
1510957418		UTF-16 BE	BOM UTF 16 BE ====> **Сибирь** 1243 <==== apfs Bl
1510961578		UTF-16	BOM UTF 16 LE ====> **Сибирь** 1242 <==== apfs HA
1510965434		UTF-8	BlueCrab UTF8 ====> **Сибирь** 1241 <==== apfs HA
1511796993		UTF-8	� P � � ко � � **Сибирь** � � �;#) (�

Two More Things Learned Making Test Data

MFT: *fixups* and the *Update Sequence Array.*

- I noticed my string documentation program sometimes missed strings that I knew were in the NTFS meta-data part of the test image, but forensic string search tools could find the strings that my program missed. See MFT fixup.

Copy/Paste from PDF may not do what you expect.

- One day I noticed that none of the tools found Arabic text anymore.
- I was copying/pasting from a PDF.
- Arabic + PDF = not quite Unicode. The string renders correctly in the search tool, but the byte codes copied are not Unicode. See the pdf spec to see why.

Coming Soon -- Federated Testing with String Search http://www.cftt.nist.gov/federated-testing.html

Sharing CFTT Test Methods, Tools & Forensic Lab Test Reports

- Helps a forensic lab test tools easily and with high quality
- For string searching CFTT provides test images with known content and a list of test cases designed to test specific features.
  1. Tester can select relevant test cases from a list of test cases
  2. Each case is run by first setting tool options and then searching for a string
  3. Federated testing tool records search results
  4. Tool to generate a skeleton test report that can then can be finished in the style favored by the laboratory.
- The test reports can be shared with other labs

A basic test case

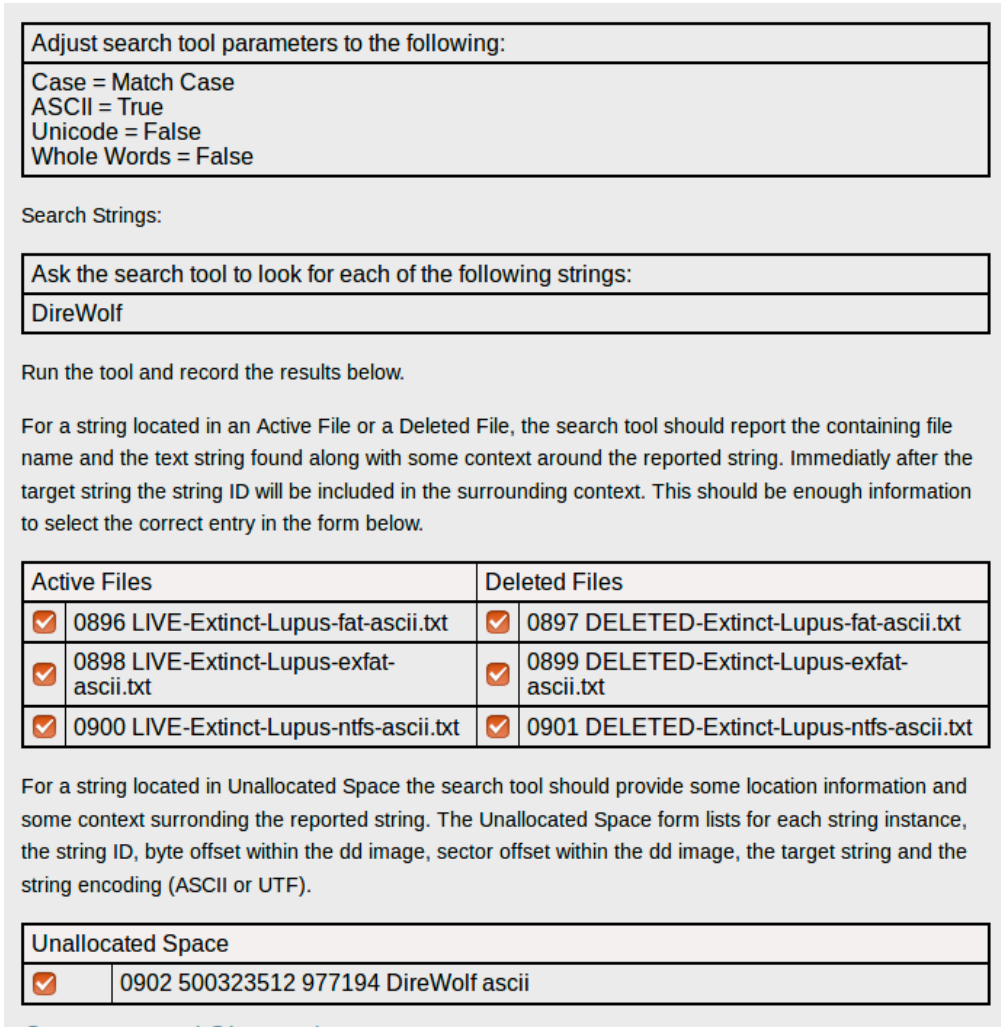
Case	Strings	Options	Case Description

FT-SS-01	DireWolf	Case = Match Case
		ASCII = True
		Unicode = False
		Whole Words = False
			Search ASCII

ID	Offset	Containing File Name
0897	8,197,307	DELETED-Extinct-Lupus-fat-ascii.txt
0896	9,172,152	LIVE-Extinct-Lupus-fat-ascii.txt
0902	500,323,512	LIVE-Extinct-Lupus-unalloc-ascii.txt
0899	1,000,839,354	DELETED-Extinct-Lupus-exfat-ascii.txt
0898	1,001,613,487	LIVE-Extinct-Lupus-exfat-ascii.txt
0900	1,504,877,750	LIVE-Extinct-Lupus-ntfs-ascii.txt
0901	1,666,325,693	DELETED-Extinct-Lupus-ntfs-ascii.txt

- Test image has 4 partitions: FAT, Unformatted, ExFAT & NTFS
- Test strings appear multiple (in this case 7) times with something different about each instance
- The search string appears twice in each formatted partition, once in unallocated space
- Each instance of the string has a unique ID, placed just after the string

Test Case Summary with Expected Results



Adjust search tool parameters to the following:

Case = Match Case
ASCII = True
Unicode = False
Whole Words = False

Search Strings:

Ask the search tool to look for each of the following strings:
DireWolf

Run the tool and record the results below.

For a string located in an Active File or a Deleted File, the search tool should report the containing file name and the text string found along with some context around the reported string. Immediatly after the target string the string ID will be included in the surrounding context. This should be enough information to select the correct entry in the form below.

Active Files		Deleted Files	
✓	0896 LIVE-Extinct-Lupus-fat-ascii.txt	✓	0897 DELETED-Extinct-Lupus-fat-ascii.txt
✓	0898 LIVE-Extinct-Lupus-exfat-ascii.txt	✓	0899 DELETED-Extinct-Lupus-exfat-ascii.txt
✓	0900 LIVE-Extinct-Lupus-ntfs-ascii.txt	✓	0901 DELETED-Extinct-Lupus-ntfs-ascii.txt

For a string located in Unallocated Space the search tool should provide some location information and some context surrounding the reported string. The Unallocated Space form lists for each string instance, the string ID, byte offset within the dd image, sector offset within the dd image, the target string and the string encoding (ASCII or UTF).

Unallocated Space	
✓	0902 500323512 977194 DireWolf ascii

- Specifies what search options to select
- Specifies what string or pattern to search for
- Presents expected results – after running the search select the checkboxes to record all strings found
- Record false hits and other notable behavior in a comment text box (not shown)

Contact Information

Jim Lyle
jlyle@nist.gov
http://www.cftt.nist.gov

E-Mail federatedtesting-request@nist.gov with the word “subscribe” (without quotes) in the subject line to subscribe to the federatedtesting@nist.gov mailing list. Federatedtesting@nist.gov is a low volume mailing list for distributing updates on the Federated Testing project and the Federated Testing Forensic Tool Testing Environment (e.g., new releases/versions and capabilities).