Signatures and Stream Utility

User Manual

V1.0

Author: Sanketh Sudharsan Iyengar

Signatures and Stream Utility

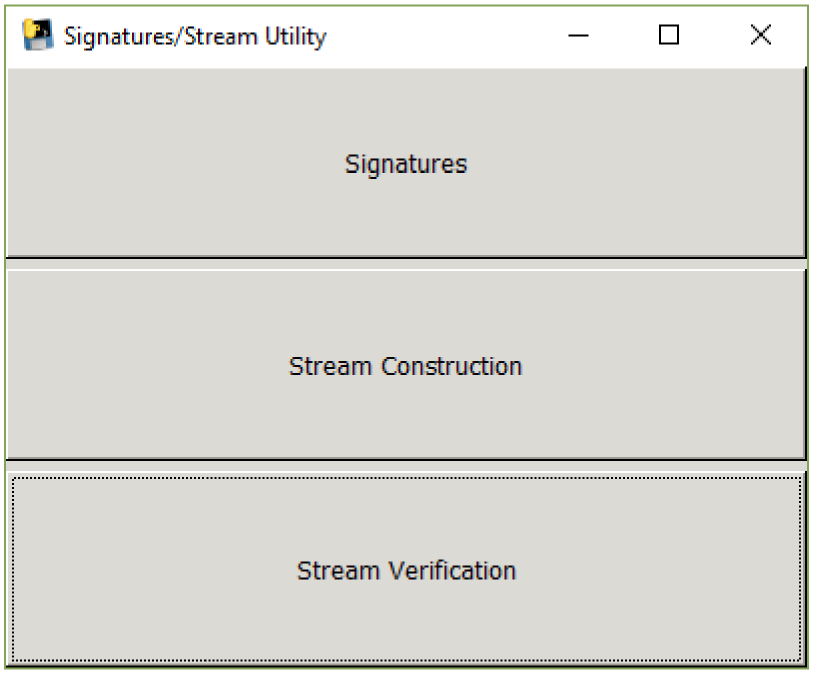
User Manual

Contents

1. UI Guide 3
1.1 Welcome Screen 3
1.2 Signatures 3
1.2.1. File 4
1.2.2. Display 4
1.2.3. Generate 5
1.2.4. Construct 5
1.3 Stream Construction 6
1.3.1. File 6
1.3.2. Display 7
1.4. Stream Verification 8
1.4.1. File 8
2. File Types 9
2.1. Hex Signatures 9
2.2. Text Signatures 9
2.3. Stream Binary File 9
2.4. Stream Map 9
2.5. Stream Stats 10
2.6. Processed Stream Stats 10
3. Code documentation 11
3.1 Constants 11
3.2 Function descriptions 11
3.3 Execution Instructions 11

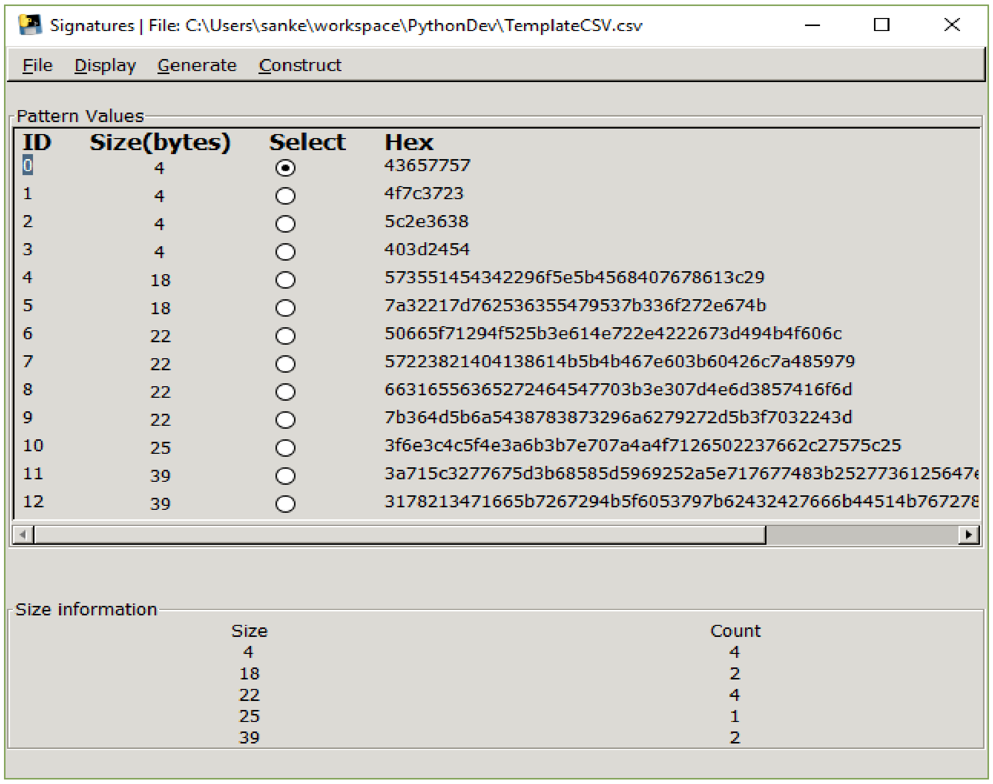
1. UI Guide

1.1 Welcome Screen



The welcome screen provides buttons to open the Signatures, Stream Construction and Stream Verification windows.

1.2 Signatures


Signatures | File: C:\Users\sanke\workspace\PythonDev\TemplateCSV.csv
File
Display
Generate
Construct

Pattern Values

ID	Size(bytes)	Select	Hex
0	4	●	43657757
1	4	○	4f7c3723
2	4	○	5c2e3638
3	4	○	403d2454
4	18	○	573551454342296f5e5b4568407678613c29
5	18	○	7a32217d762536355479537b336f272e674b
6	22	○	50665f71294f525b3e614e722e4222673d494b4f606c
7	22	○	57223821404138614b5b4b467e603b60426c7a485979
8	22	○	66316556365272464547703b3e307d4e6d3857416f6d
9	22	○	7b364d5b6a5438783873296a6279272d5b3f7032243d
10	25	○	3f6e3c4c5f4e3a6b3b7e707a4a4f7126502237662c27575c25
11	39	○	3a715c3277675d3b68585d5969252a5e717677483b2527736125647
12	39	○	3178213471665b7267294b5f6053797b62432427666b44514b76727

Size information

Size	Count
4	4
18	2
22	4
25	1
39	2

1.2.1. File

1.2.1.1 Load Hex Signatures with IDs and Size(.csv)

Input File Type: Refer section 2.1.

Loads Hex signatures with IDs and Size in the form of a CSV File. Populates the GUI with the given data.

1.2.1.2 Load Text Signatures(.txt)

Input File Type: Refer section 2.2.

Loads text signatures the form of a text File. It automatically generates a signature ID and creates its Hex equivalent. Populates the GUI with the given data.

1.2.1.3 Save As

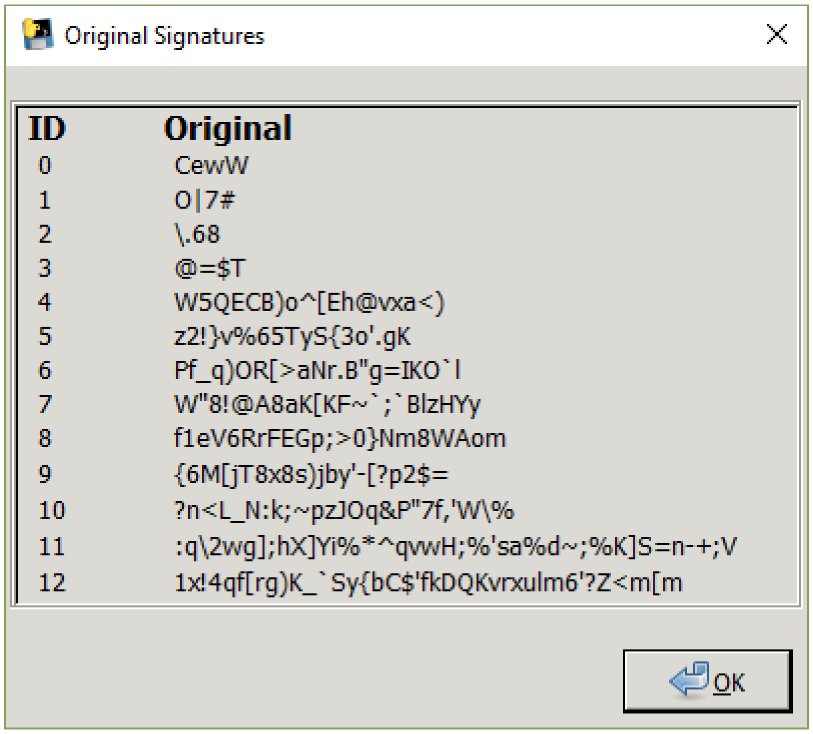
Output File Type: Refer section 2.1.

Saves the signature data as a CSV file with ID, Size and Hex values for each signature.

1.2.2. Display

1.2.2.1. Display Text View of Signatures

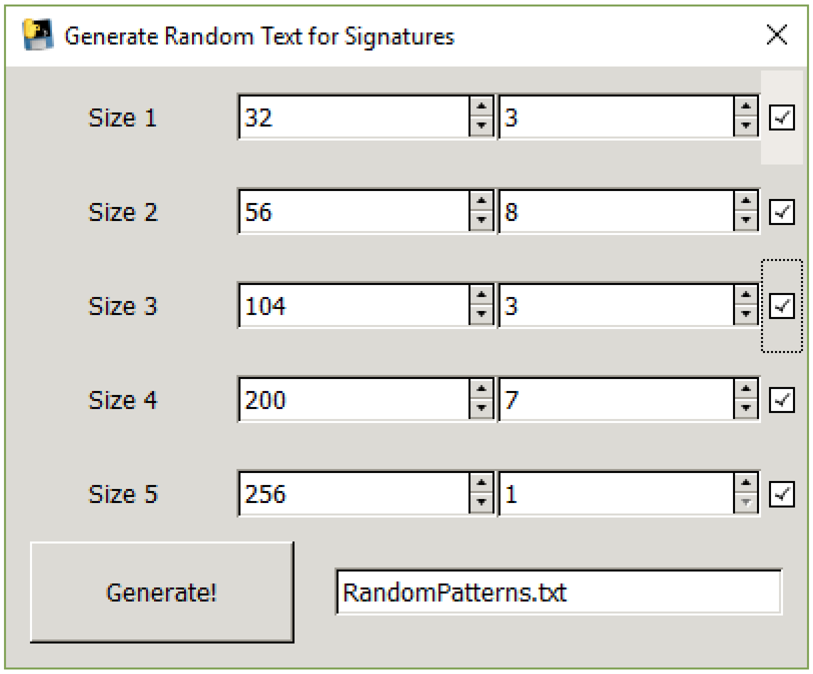
Displays the Signatures as plain text. Hex data is reconverted to string and displayed on the GUI.



```
ID        Original
 0         CewW
 1         O|7#
 2         \.68
 3         @=$T
 4         W5QECB)o^[Eh@vxa<)
 5         z2!}v%65TyS{3o'.gK
 6         Pf_q)OR[>aNr.B"g=IKO`l
 7         W"8!@A8aK[KF~`;`BlzHYy
 8         f1eV6RrFEGp;>0}Nm8WAom
 9         {6M[jT8x8s)jby'-[?p2$=
10         ?n<L_N:k;~pzJOq&P"7f,'W\%
11         :q\2wg];hX]Yi%*^qvwH;%'sa%d~;%K]S=n-+;V
12         1x!4qf[rg)K_`Sy{bC$'fkDQKvrxulm6'?Z<m[m
```

1.2.3. Generate

1.2.3.1. Generate Random Text Signatures



Size 1	32	3	✓
Size 2	56	8	✓
Size 3	104	3	✓
Size 4	200	7	✓
Size 5	256	1	✓

RandomPatterns.txt

Output File Type: Refer section 2.2.

Opens a dialog box to help generate random text signatures. Tooltips describe the purpose of each input location. First column is to specify the size (in bits), second column is to specify the count of each size, and the last checkbox is to enable creation for the specified combination.

File name is entered in the entry box and on clicking generate, a text file is created **in the local directory**. This can be repeated as many times as required with different combinations.

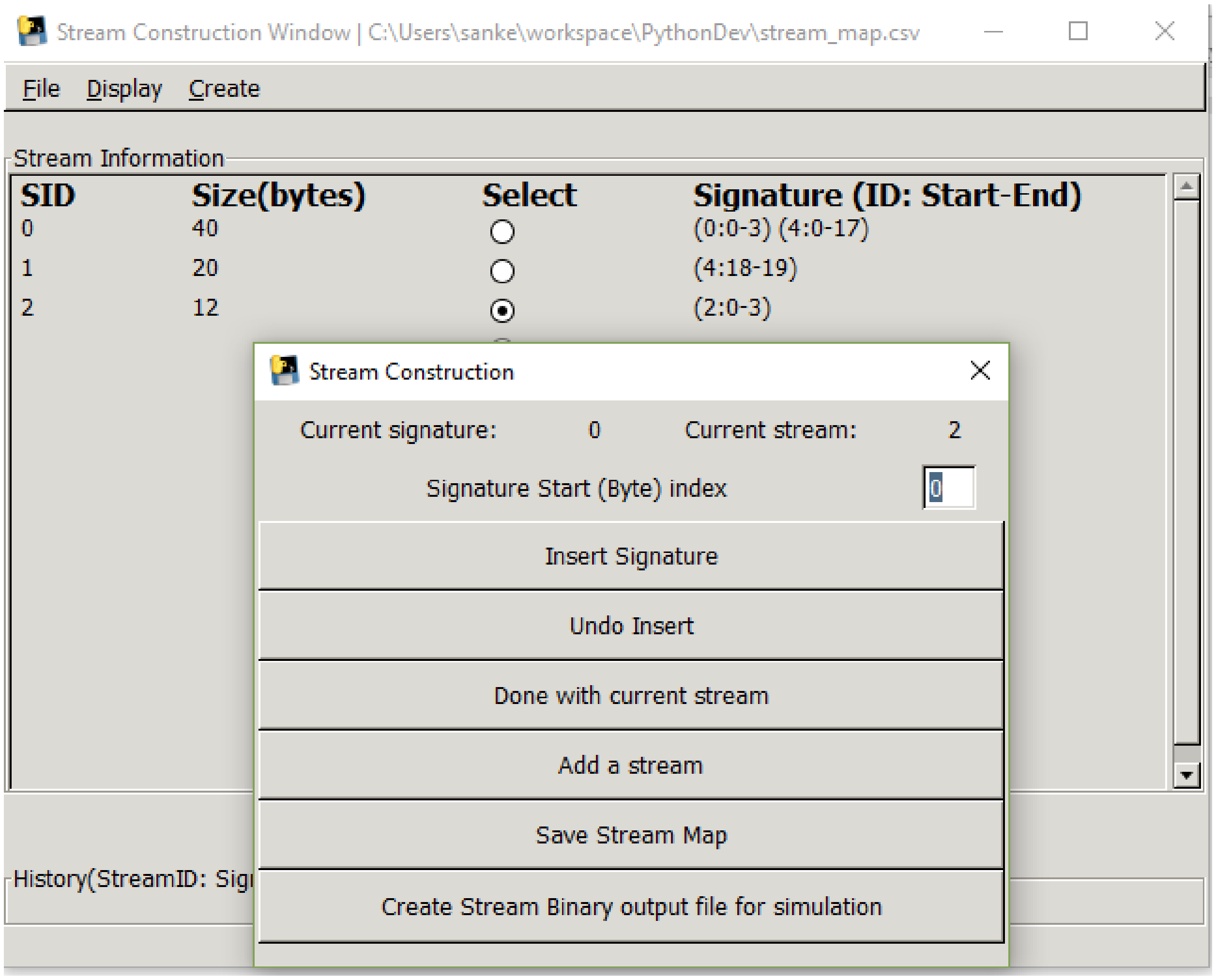
The generated file can be loaded using File->Load Text Signatures(.txt).

1.2.4. Construct

1.2.4.1. Construct Stream

Opens the Stream construction window. Select lines in the signature window are enabled on pressing this option.

1.3 Stream Construction



SID	Size(bytes)	Select	Signature (ID: Start-End)
0	40	○	(0:0-3) (4:0-17)
1	20	○	(4:18-19)
2	12	●	(2:0-3)

Current signature: 0 Current stream: 2

Signature Start (Byte) index 0

1.3.1. File

1.3.1.1. Load Stream Binary File

Input File Type: Refer section 2.3.

Loads a stream binary File and populates the GUI. It also opens the Stream Construction Dialog for modifying the loaded streams. Undo option is disabled as there is no way of retrieving the original stream.

1.3.1.2. Load Stream Map

Input File Type: Refer section 2.4.

Loads a stream map File and populates the GUI. It also opens the Stream Construction Dialog for modifying the stream map. Final streams are created only when 'Create Stream Binary output file for simulation' is selected in the dialog.

1.3.2. Display

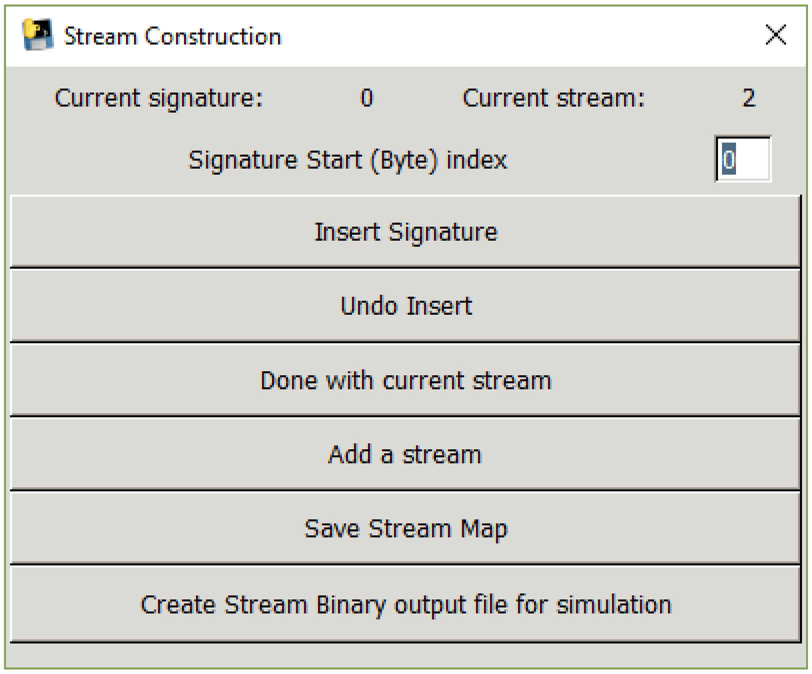
1.3.2.1. Display Text View of Current Stream

Displays the current Stream as plain text.

Note: Since signatures are inserted only when Binary File is created, this will show the streams without signatures being applied.

1.3.3. Create

1.3.3.1. Open Stream Construction Dialog



Current signature: Displays the active signature. Select a new radio button in the Signature Window to update the selection.

Current stream: Displays the active stream. Select a new radio button in the Stream Window to update the selection.

Signature Start (Byte) index: Starting byte index for inserting a signature into a stream

Insert Signature: Inserts the Signature in the location specified by 'Signature Start (Byte) index'. Generates a warning/error message in case of incorrect insertion.

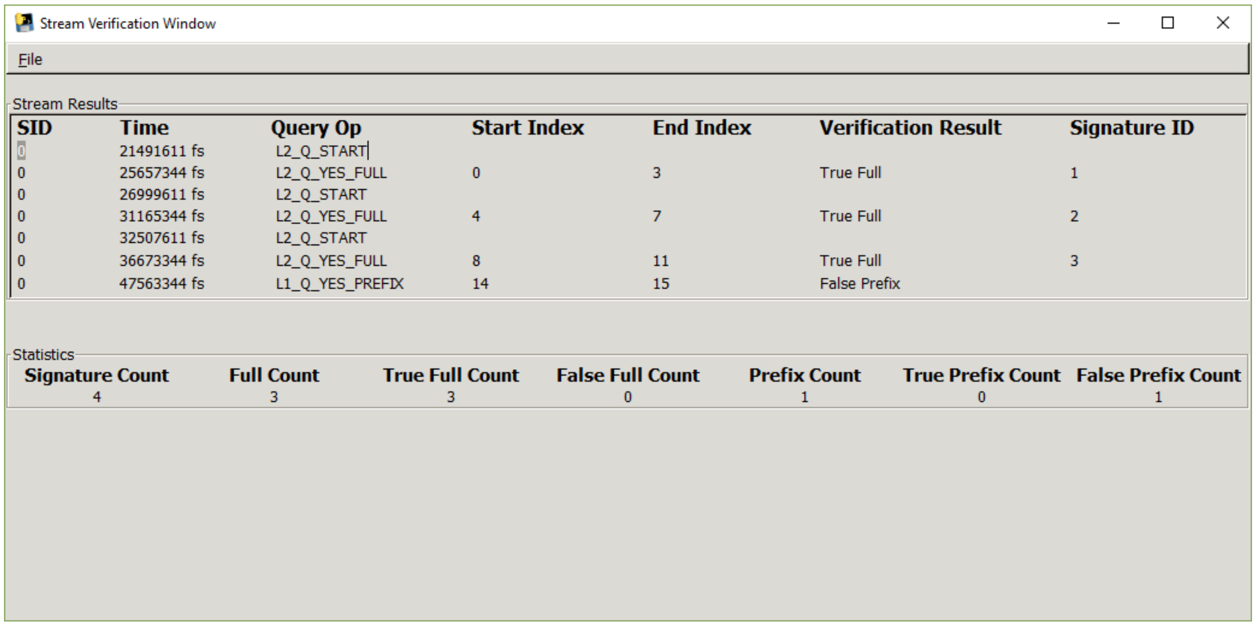
Undo Insert: Removes the last inserted signature from the map

Done with current stream: Moves the active stream to the next stream.

Add a stream: Opens a dialog for adding new stream. Stream ID and Stream Size are the required inputs.

Create Stream Binary output file for simulation: Creates a binary file as described in section 2.3, to be loaded as stream input for the simulator.

1.4. Stream Verification



Stream Results

SID	Time	Query Op	Start Index	End Index	Verification Result	Signature ID
0	21491611 fs	L2_Q_START				
0	25657344 fs	L2_Q_YES_FULL	0	3	True Full	1
0	26999611 fs	L2_Q_START				
0	31165344 fs	L2_Q_YES_FULL	4	7	True Full	2
0	32507611 fs	L2_Q_START				
0	36673344 fs	L2_Q_YES_FULL	8	11	True Full	3
0	47563344 fs	L1_Q_YES_PREFIX	14	15	False Prefix	

Statistics

Signature Count	Full Count	True Full Count	False Full Count	Prefix Count	True Prefix Count	False Prefix Count
4	3	3	0	1	0	1

1.4.1. File

1.4.1.1. Load Stream Stats

Input File Type: Refer section 2.5.

Loads Stream Stats into the window and generates statistics for True/False Full/Prefix signatures.

1.4.1.2. Load processed Stream Stats

Input File Type: Refer section 2.6.

Loads previously processed Stream Stats into the window.

1.4.1.3. Save processed Stream Stats

Output File Type: Refer section 2.6.

Saves Stream Stats and statistics in CSV format.

2. File Types

2.1. Hex Signatures

Sample File: /Sample Files/HexSignatures.csv

CSV file with columns ID (Signature ID), Size (in bits) and Hex signatures.

```
0,32,43657757
1,32,4f7c3723
2,32,5c2e3638
3,32,403d2454
4,144,573551454342296f5e5b4568407678613c29
```

2.2. Text Signatures

Sample File: /Sample Files/TextSignatures.txt

Text file with string signatures.

```
Zojt>Gmt94VYry)W0=ZK?TJG|&6A*\bn"9:3kc6
%bwy$VdO#9[VdxF4,.)AS|!w-pY2:ivbFX*i-@)rybnS',,C]Q7^j6Wsp(&l6HO?T;9)0+`4/xwf/bGj|;)LW{>L'Hj+XGXFL\AG5cpPzyi.
;Jt$}==_Q(x+\n#|718i:atin;%~bg,#[G[xaIj9BFv\<%(Pbo35NWAl7bnh'GnqK@L[c,QViSU[i7f\`=t:\RRQ+|WFqkqacK,:|_]+'vG<<5}
{nrYLc<;6F.Q8)XN7_`n8ZEk0jgDlyNC9D+DL6U&cs|0/o*|k{T)+Rk>yt-9_\,ltFToy=vQkO?Em}W*w**5j%Jckz=dDw;TU+fA`<]d0Rau\i=~]:
G_QkqYm~kJ-#|GUZr?%T~3i>[?T
```

2.3. Stream Binary File

Sample File: /Sample Files/StreamBinary.csv

CSV file with columns SID (Stream ID), Stream Size (in bits), Hex streams, Signature Count and Signature Information. Signature information is of the form: Signature ID, Start Index, End Index.

First line of this file is the Load Line with values in the format: L, SLR0, SLR1, SLR2, SLR3, SLR4. SLR values are filled based on the priority of 1) SLR0 for holds the largest 2) SLR4 holds the smallest 3) After SLR0 is filled, based on number of signature lengths SLR4, SLR3, SLR3, SLR1 are filled based on the size of signatures.

```
L,88,0,0,0,0
1,112,746573745f7369676e5f322c4754,1,1,0,10
```

2.4. Stream Map

Sample File: /Sample Files/StreamMap.csv

CSV file with columns SID (Stream ID), Stream Size (in bits), ID (Signature ID), Start Index, End Index.

Each Line represents a new signature. In case there are no signatures in a stream, ID, Start Index and End Index are ALL equal to 0.

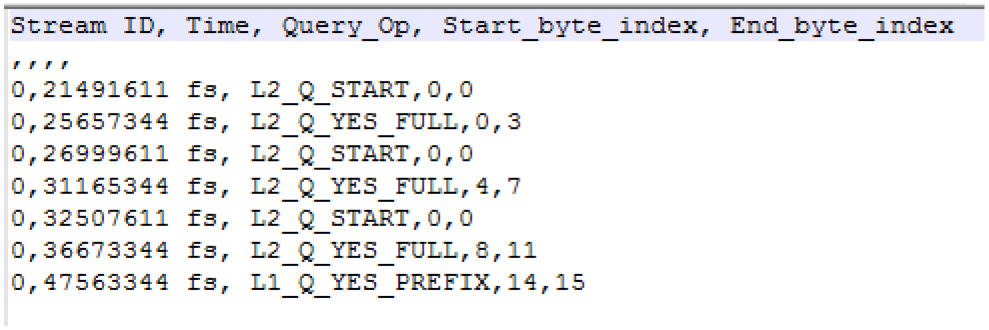
```
0,320,0,0,3,1
0,320,4,0,17,1
1,160,4,18,19,1
2,96,2,0,3,1
```

2.5. Stream Stats

Sample File: /Sample Files/StreamStats.csv

CSV file with columns Stream ID, Time, Query Operation, Start byte index and End byte index.

Note: Second line is expected to be blank while loading.



```
Stream ID, Time, Query_Op, Start_byte_index, End_byte_index
,,,,
0,21491611 fs, L2_Q_START,0,0
0,25657344 fs, L2_Q_YES_FULL,0,3
0,26999611 fs, L2_Q_START,0,0
0,31165344 fs, L2_Q_YES_FULL,4,7
0,32507611 fs, L2_Q_START,0,0
0,36673344 fs, L2_Q_YES_FULL,8,11
0,47563344 fs, L1_Q_YES_PREFIX,14,15
```

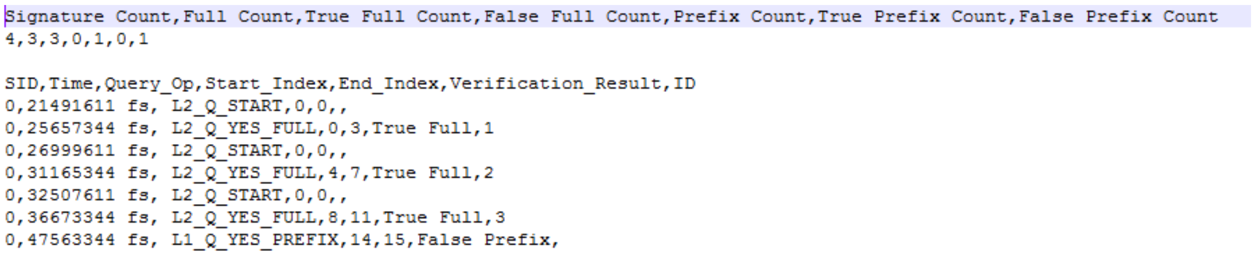
2.6. Processed Stream Stats

Sample File: /Sample Files/ProcessedStreamStats.csv

Two-part CSV file with first part containing statistics summary information and second part containing the statistics.

Statistics summary information has the following columns: Signature Count, Full Count, True Full Count, False Full Count, Prefix Count, True Prefix Count and False Prefix Count.

Statistics has the following columns SID (Stream ID), Time, Query Operation, Start index, End index, Verification Result and ID (Signature ID).



```
Signature Count,Full Count,True Full Count,False Full Count,Prefix Count,True Prefix Count,False Prefix Count
4,3,3,0,1,0,1

SID,Time,Query_Op,Start_Index,End_Index,Verification_Result,ID
0,21491611 fs, L2_Q_START,0,0,,
0,25657344 fs, L2_Q_YES_FULL,0,3,True Full,1
0,26999611 fs, L2_Q_START,0,0,,
0,31165344 fs, L2_Q_YES_FULL,4,7,True Full,2
0,32507611 fs, L2_Q_START,0,0,,
0,36673344 fs, L2_Q_YES_FULL,8,11,True Full,3
0,47563344 fs, L1_Q_YES_PREFIX,14,15,False Prefix,
```

3. Code documentation

3.1 Constants

1) Constant to change signatures utility name
```
TOOL_NAME = "Signatures"
```

2) Constants to determine number of signature length
```
MIN_N_LENGTHS = 1
MAX_N_LENGTHS = 5
```

3) Constants to customize allowable signature length
```
MIN_SIZE_SIGNATURE = 4*8
MAX_SIZE_SIGNATURE = 130*8
DISTANCE_CONSECUTIVE = 2*8
```

4) Allowable constants for random signatures and streams
```
ALLOWABLE_CHARS = string.ascii_letters + string.digits + string.punctuation
```

5) Maximum number of streams being processed at a time
```
MAX_COUNT_STREAMS = 15
```

3.2 Function descriptions

Please refer to /Source/Signatures_Stream_Utility.py for individual function comments.

3.3 Execution Instructions

Run the code using the following two options:

1) Using Source file
   a) Open a command prompt window in the directory of the source file
   b) Type python Signatures_Stream_Utility.py
2) Using Executable file
   a) Browse to /Executable/Packaged or /Executable/Stand-alone folder
   b) Run Signatures_Stream_Utility.exe

Note: Stand-alone exe execution will be slower, since it needs to unpack before execution.