

**/proc/meminfo Covert Channel Operation**  
**Written by: Benjamin Steenkamer, February 2018**

**Relevant Source Variables:**

MemFree: Amount of system memory in kB currently unused by the system

M: Number of start sequence bits to be transmitted by source (pre-established between source and sink)

N: Number of data bits to be transmitted by source

D<sub>start</sub>: Vector containing the M start sequence bits (pre-established between source and sink)

D<sub>data</sub>: Vector containing the N data bits

HIGH\_BIT\_ALLOC: Predefined amount of memory in kB that will be allocated to represent a 1

LOW\_BIT\_ALLOC: Predefined amount of memory in kB that will be allocated to represent a 0

**Source Operations:**

Start source program;

**for** CALIB\_TIME seconds **do**

Record the value of MemFree every CALIB\_DELAY microseconds;

**end for**

Calculate average MemFree value;

**if** system doesn't have enough free memory **then**

End source program;

**end if**

**for** i = 0 to M - 1 **do**

**if** D<sub>start</sub>[i] == 1 **then**

Allocate HIGH\_BIT\_ALLOC amount of memory;

**else then**

Allocate LOW\_BIT\_ALLOC amount of memory;

**end if**

Set memory chunk so system will decrement FreeMem value;

Sleep for HOLD\_TIME microseconds;

Free allocated memory so system will increment FreeMem value;

Sleep for HOLD\_TIME microseconds;

**end for**

**for** i = 0 to N - 1 **do**

**if** D<sub>data</sub>[i] == 1 **then**

Allocate HIGH\_BIT\_ALLOC amount of memory;

**else then**

Allocate LOW\_BIT\_ALLOC amount of memory;

Set memory chunk;

Sleep for HOLD\_TIME microseconds;

Free allocated memory;

Sleep for HOLD\_TIME microseconds;

**end if**

**end for**

End source program;

**Relevant Sink Variables:**

MemFree: Amount of system memory in kB currently unused by the system

M: Number of start sequence bits to be transmitted by source (pre-established between source and sink)

$D_{start}$ : Vector containing the M start sequence bits (pre-established between source and sink)

$D_{recordings}$ : The MemFree values recorded by the sink

$D_{data}$ : 1's and 0's derived from the recorded MemFree values

ZERO\_UPPER\_LIMIT: Maximum value of MemFree that will be recognized as 0

ONE\_UPPER\_LIMIT: Maximum value of MemFree that will be recognized as 1

(ONE\_UPPER\_LIMIT < ZERO\_UPPER\_LIMIT < average MemFree value)

**Sink Operations:**

Start sink program;

**for** CALIB\_TIME seconds **do**

Record the value of MemFree every CALIB\_DELAY microseconds;

**end for**

Calculate average MemFree value;

Using average and predefined values, calculate ONE\_UPPER\_LIMIT and ZERO\_UPPER\_LIMIT;

**if** system doesn't have enough free memory **then**

End sink program;

**end if**

**for** CHANNEL\_TIME seconds **do**:

Record the value of MemFree every RECORD\_DELAY microseconds;

**end for**

**for** i = 0 to length( $D_{recordings}$ ) - 1 **do**

**if**  $D_{recordings}[i]$  is a local minimum **and**  $D_{recordings}[i] \leq ONE\_UPPER\_LIMIT$  **then**

Append 1 to  $D_{data}$ ;

**else if**  $D_{recordings}[i]$  is a local minimum **and**  $D_{recordings}[i] \leq ZERO\_UPPER\_LIMIT$  **then**

Append 0 to  $D_{data}$ ;

**end if**

**end for**

**for** i = 0 to length( $D_{data}$ ) - M **do**

// If a subsequence of  $D_{data}$  equals the start sequence,  $D_{start}$

**if**  $\langle D_{data}[i], \dots, D_{data}[i + \text{length}(D_{start}) - 1] \rangle == D_{start}$  **then**

start\_sequence\_found = true;

data\_index = i + length( $D_{start}$ );

break;

**end if**

**end for**

**if** start\_sequence\_found == true **then**

Starting at data\_index, display rest of values in  $D_{data}$ ;

**end if**

End sink program;