System Tracer Tool: Design and Operation Document

By Liam Pilling, Timothy Covich, and Jacob Boland

Abstract

This document outlines the goal, design, and usage instructions for the system tracer tool developed.

Table of Contents

Purpose

The purpose of this tool is to monitor the operation of a Windows or Linux system and provide detailed information to the user. It runs in the background, recording all system calls made and software libraries used during a specified time interval. This information is then output to file in a readable format for review by the user. This will allow for the identification of errant or unexpected activity and allow users to take action to rectify the issue.

Acknowledgements

In the development of this tool we would like to acknowledge the use of external software and tools and the authors of said software. Python, is used as the underlying platform for both the Windows and Linux distributions. Procmon is a Windows process monitoring tool used to retrieve information on running processes and the activities of said processes. Listdlls, as the name implies, lists all the dlls in use on a Windows system and is used to retrieve that information to link them to the processes that call them. Strace is a command line tool for Linux systems that monitors a single process and records the system calls made. This tool is used to monitor all processes running on a Linux system and extract the information to a readable format.

System Requirements

Windows

- A system running Windows XP x32, Windows 7 x64, or Windows 10 x64
- Python 3.6 or 3.7
- Microsoft Visual Redistributable Package
- psutil – a required Python module not included by default
- pandas – a required Python module not included by default
- Administrative privileges

Linux

- Theoretically any Linux distribution able to run python 3, strace and the timeout console command but it has only been tested on an Ubuntu 18.4 system
- Python 3
- Strace
- Timeout 8.28
- Administrative privileges

Components and Design

This tool is designed to capture the system calls and DLL usage made during a given period of time on both Windows and Linux systems. The code consists of two separate components, the Linux code package and the Windows package, as well as a base launcher. The base launcher, 'multiplayer.py', is a simple python script used to determine whether the tool is being run on a Windows or Linux system and then calls the appropriate package.

All components of the Windows distribution are found in the win 32 folder and consists of 'tracermon.py', 'Procmon.exe', 'config.pmc', and 'Listdlls.exe'. The main controller of the Windows distribution is 'tracermon.py' which runs the program, sorts the output, and performs the statistical analysis. 'Procmon.exe', otherwise known as Windows Process Monitor, is run in a minimal mode, as opposed to the usual GUI operation, and is used to capture all the system calls made by the system.

This tool was found to be the most efficient way to gather this information with solutions such as NTtrace, straceNT, and DrMemory either not working, crashing, or tracing only one process while causing the machine to hang until finished. Furthermore, due to time constraints it was unreasonable to build a tool that was able to capture all of the desired information from the ground up but with more time it may be possible. It also requires the configuration file 'config.pmc' for its operation. The 'Listdll.exe' executable is a command line program also written by Microsoft and SysInternals. It outputs all the DLLs being used on the Windows system it is operating on and the output is extracted by 'tracemon.py' for display and statistical analysis. Unfortunately, due to a misstep in the design it has been found that 'Listdlls.exe' only checks DLL usage at the start of the applications runtime. This means that it may miss DLLs used in an extended run but due to time constraints there was no way to find an alternate solution.

The Linux distribution consists entirely of 'tracer.py' utilising the Strace tool to monitor the calls made by each process during the runtime. It first ensures that the program is being run with suitable privileges and that strace is installed, both of which are required for proper operation. It then creates a list of all pids, finds all the shared objects of the processes by grabbing the linked shared objects from the map files in the proc folder, and builds the command to trace all the processes based on the time interval provided. The program then runs strace on all process ID's it has found and any child processes they spawn during the runtime and then waits for the specified runtime. After completion, the raw data is analysed, compiled into a human readable format, and output to 'TraceCallSummary.txt' and 'MapSummary.txt'.

Installation and Usage

Windows

Installation

1. Install Python 3.7: Included as part of the package is the 'python-3.7.1-amd64.exe' file. This should be installed if your system does not already have Python 3 installed as seen in figure 1.1. Unless you wish to customise your path or alter the options, use the 'Install Now' feature. It is also recommended you check the 'Add Python 3.7 to PATH' for easier use.
   Please note that the executable required is for x64 versions of Windows and is not suitable for x86 variants. If you are using an x86 version please visit https://www.python.org/downloads/ for more information.
2. If your system does not have the Microsoft Visual Redistributable Package installed the error seen in figure 1.2 will occur when trying to run Python. To fix this, run the included "vc_redist.x86.exe" included as part of the package.
3. The tool also requires the installation of the "psutil" and "pandas" Python modules. From the Windows command line interface run:

   ```
   > python – m pip install psutil
   > python -m pip install pandas
   ```

   If successful, the output should be similar to figure 1.3. If, upon attempting to run the tool, an error similar to figure 1.4 is shown it means the installation of the "psutil" or "pandas" module was unsuccessful and will need to be repeated.

Usage

1. Open a Windows Command Prompt in the project folder ensuring that it is run as administrator.
2. Run multiplayer.py

   a. If your python 3 executable path was set during installation simply use ‘python multiplayer.py’ or ‘python3 multiplayer.py’. The difference is dependent on your individual PATH setup.
   b. If the path was not set during installation use you must point toward the python 3 executable manually. For example, ‘“C:\Program Files x86\Python 3\python.exe” multiplayer.py’. The location required may differ if you altered the installation location during setup.
   c. If you wish to provide a custom run-time use ‘python multiplayer.py 30’. This will set the program to run for 30 seconds. The time interval can be altered if desired.
3. Once the application has finished, as seen in figure 2.1, the information captured can be found in the output files which are found in the win32\working folder. Inside the working folder they are further sorted into folders in an ISO 8601-style standard (YEAR-MONTH-DAY-HOUR-MINUTE-SECOND) making it easy to find the latest output.
4. Each folder contains the files ‘raw.pml’, ‘output.csv’, and ‘statistics.txt’.
   a. The file ‘raw.pml’ contains the unreadable raw data which is captured by the tool
   b. ‘output.csv’ separates the captured data into a comma separated value style output as seen in figure 2.2. The columns are separated into ‘Date & Time’, ‘PID’, ‘Process Name’, ‘Operation’, ‘Command Line’, ‘Result’, ‘Detail’, and ‘User’
   c. The ‘statistics.txt’ file separates each process that has been monitored, lists the system calls it made, the number of times it made that call, and the DLLs it used. An example of this can be seen in figure 2.3 and 2.4.

Linux

Installation

1. Install Python 3.7 with ‘sudo apt-get install python3.7’ as seen in figure 3.1
2. Install strace with ‘apt-get install strace’
3. Ensure the timeout command from the coreutils package is available on your system.

Usage

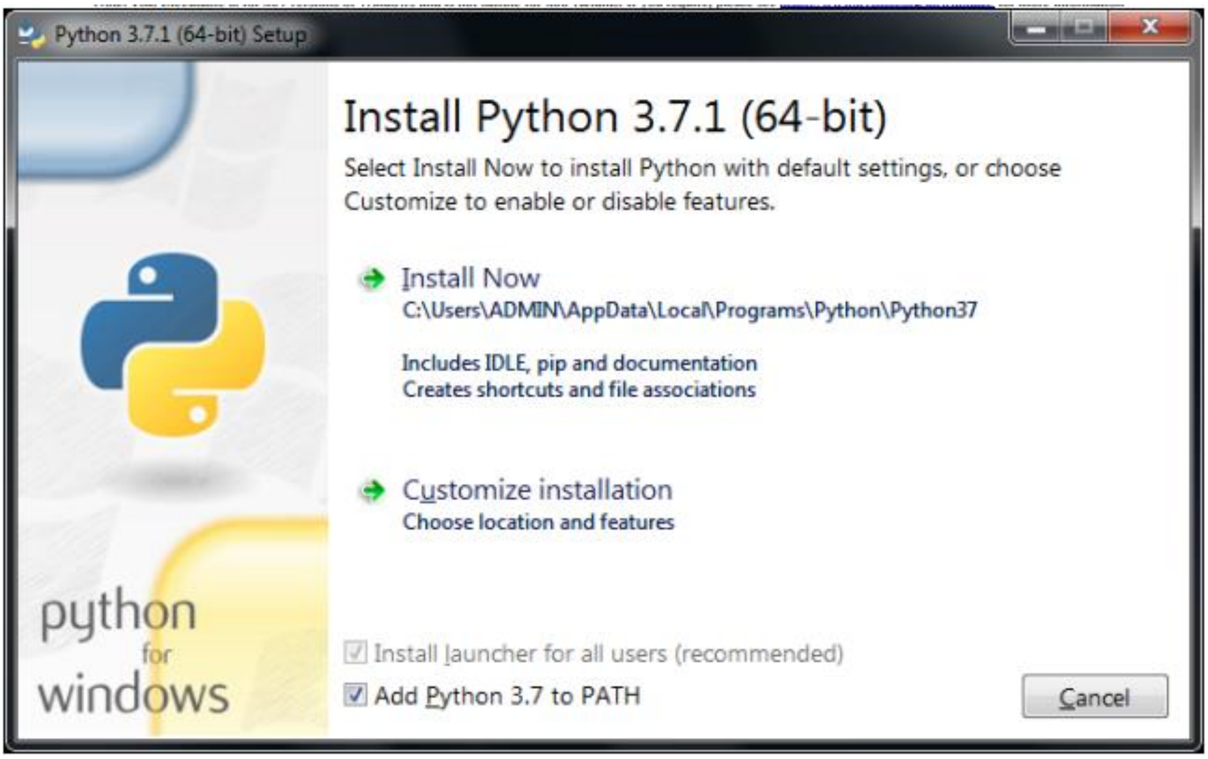
1. Open a terminal console
2. Use the command ‘sudo multiplayer.py [TIME]’ where [TIME] is the desired runtime in seconds.
3. Once the program has terminated the output can be found in ‘TraceCallSummary.txt’. An example of the output can be seen in figure 4.1.

Known Issues

- On Windows 10 machines the ListDLLs process may hang on occasion for an unknown reason. An example of this can be seen in figure 5.1. The executable “debug-killme.exe” can be used to stop the process if this occurs.
- As seen in figure 5.2 Windows systems occasionally warns that part of the output is corrupt. However, this does not seem to have any actual effect on the output or the program running
- On a fresh machine install there is a chance that a license agreement popup may appear on Windows machines as seen in figure 5.3 despite the use of the “/accepteula” command-line switch. Press “Accept” to continue with operation.
- Due to a misstep in the design the DLLusage is only being checked at the start of the applications runtime instead of throughout its operation as initially planned on the Windows version.

- The statistics output is not currently sorted by PID and is instead ordered by the order it appears in the 'raw.pml' and 'output.csv' files.
- If the Linux program crashes before completing the created temp files will not be deleted.

Appendix



Python 3.7.1 (64-bit) Setup

Install Python 3.7.1 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

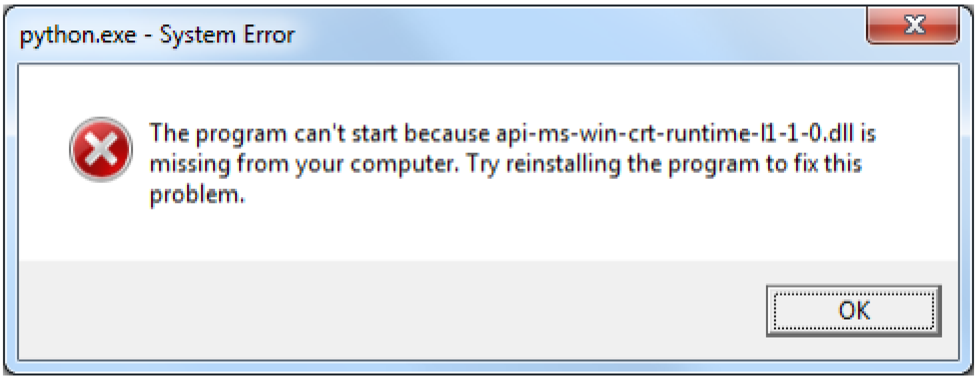
Install Now
C:\Users\ADMIN\AppData\Local\Programs\Python\Python37

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH

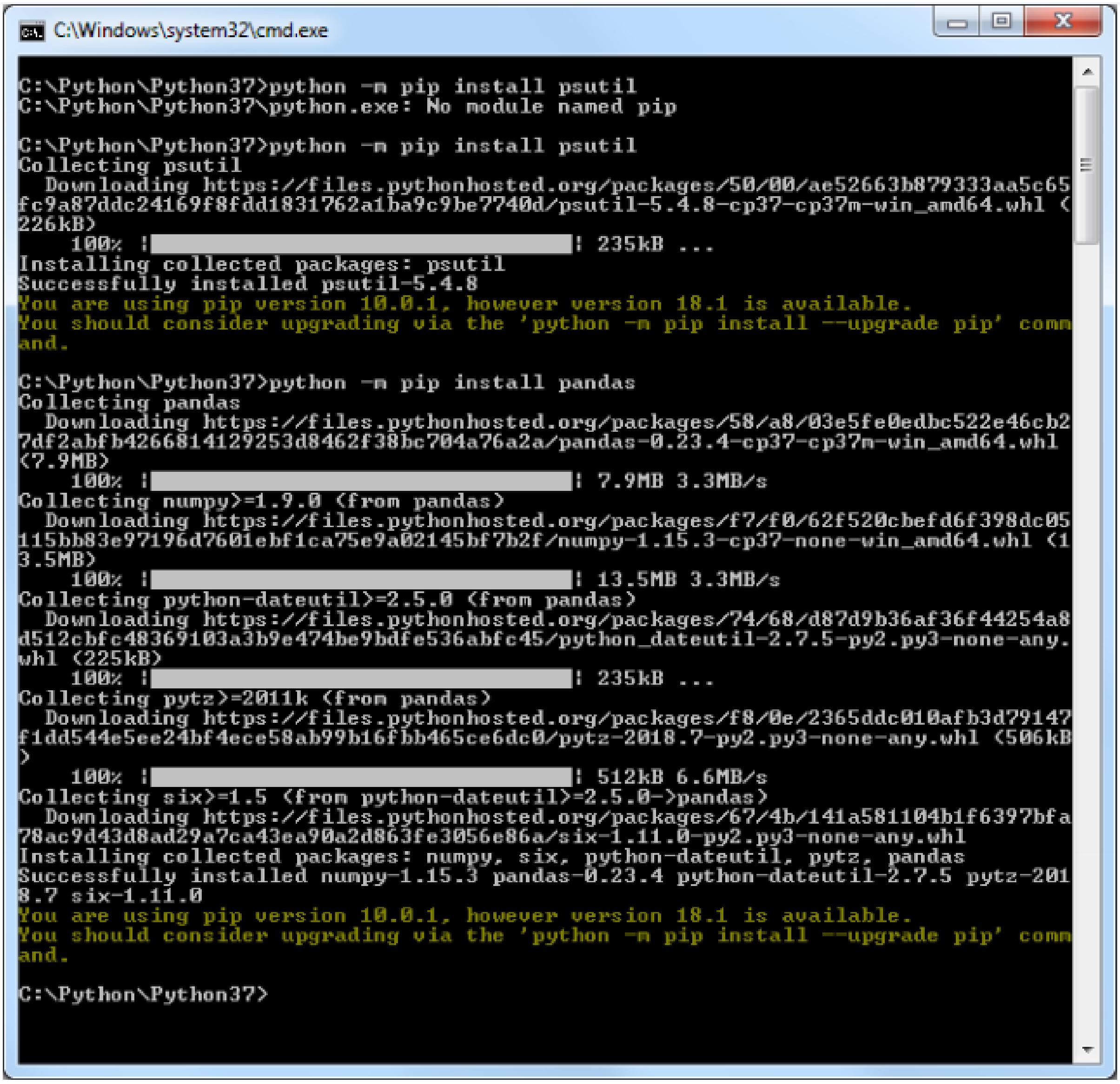
Figure 1.1: Windows x64 installer for Python 3.7



python.exe - System Error

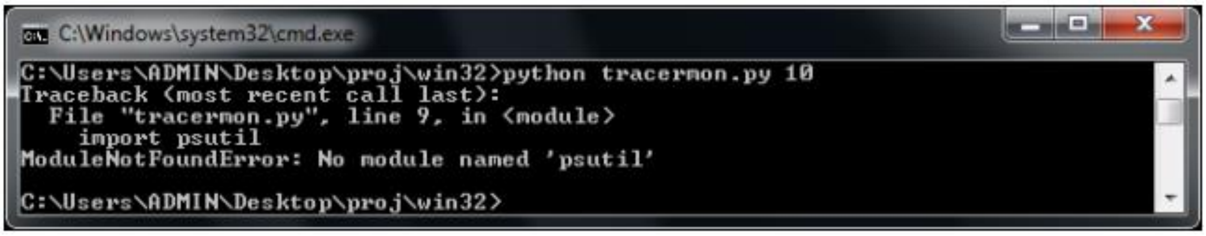
The program can't start because api-ms-win-crt-runtime-l1-1-0.dll is missing from your computer. Try reinstalling the program to fix this problem.

Figure 1.2: Error message that occurs when attempting to run Python without the Microsoft Visual Redistributable Package installed



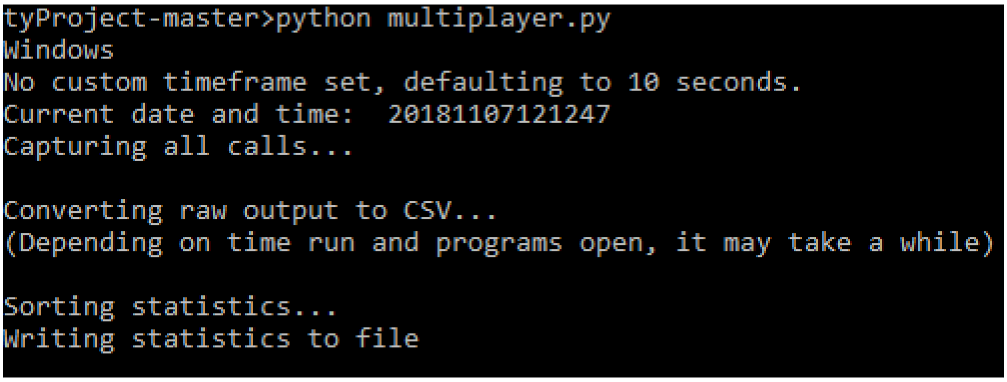
```
C:\Python\Python37>python -m pip install psutil
C:\Python\Python37\python.exe: No module named pip

C:\Python\Python37>python -m pip install psutil
Collecting psutil
  Downloading https://files.pythonhosted.org/packages/50/00/ae52663b879333aa5c65
fc9a87ddc24169f8fdd1831762a1ba9c9be7740d/psutil-5.4.8-cp37-cp37m-win_amd64.whl (
226kB)
    100% |████████████████████████████████| 235kB ...
Installing collected packages: psutil
Successfully installed psutil-5.4.8
You are using pip version 10.0.1, however version 18.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' comm
and.

C:\Python\Python37>python -m pip install pandas
Collecting pandas
  Downloading https://files.pythonhosted.org/packages/58/a8/03e5fe0edbc522e46cb2
7df2abfb4266814129253d8462f38bc704a76a2a/pandas-0.23.4-cp37-cp37m-win_amd64.whl
(7.9MB)
    100% |████████████████████████████████| 7.9MB 3.3MB/s
Collecting numpy>=1.9.0 (from pandas)
  Downloading https://files.pythonhosted.org/packages/f7/f0/62f520cbefd6f398dc05
115bb83e97196d7601ebf1ca75e9a02145bf7b2f/numpy-1.15.3-cp37-none-win_amd64.whl (1
3.5MB)
    100% |████████████████████████████████| 13.5MB 3.3MB/s
Collecting python-dateutil>=2.5.0 (from pandas)
  Downloading https://files.pythonhosted.org/packages/74/68/d87d9b36af36f44254a8
d512cbfc48369103a3b9e474be9bdfe536abfc45/python_dateutil-2.7.5-py2.py3-none-any.
whl (225kB)
    100% |████████████████████████████████| 235kB ...
Collecting pytz>=2011k (from pandas)
  Downloading https://files.pythonhosted.org/packages/f8/0e/2365ddc010afb3d79147
f1dd544e5ee24bf4ece58ab99b16fbb465ce6dc0/pytz-2018.7-py2.py3-none-any.whl (506kB
)
    100% |████████████████████████████████| 512kB 6.6MB/s
Collecting six>=1.5 (from python-dateutil>=2.5.0->pandas)
  Downloading https://files.pythonhosted.org/packages/67/4b/141a581104b1f6397bfa
78ac9d43d8ad29a7ca43ea90a2d863fe3056e86a/six-1.11.0-py2.py3-none-any.whl
Installing collected packages: numpy, six, python-dateutil, pytz, pandas
Successfully installed numpy-1.15.3 pandas-0.23.4 python-dateutil-2.7.5 pytz-201
8.7 six-1.11.0
You are using pip version 10.0.1, however version 18.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' comm
and.

C:\Python\Python37>
```

Figure 1.3: Successful installation of the psutil and pandas Python modules



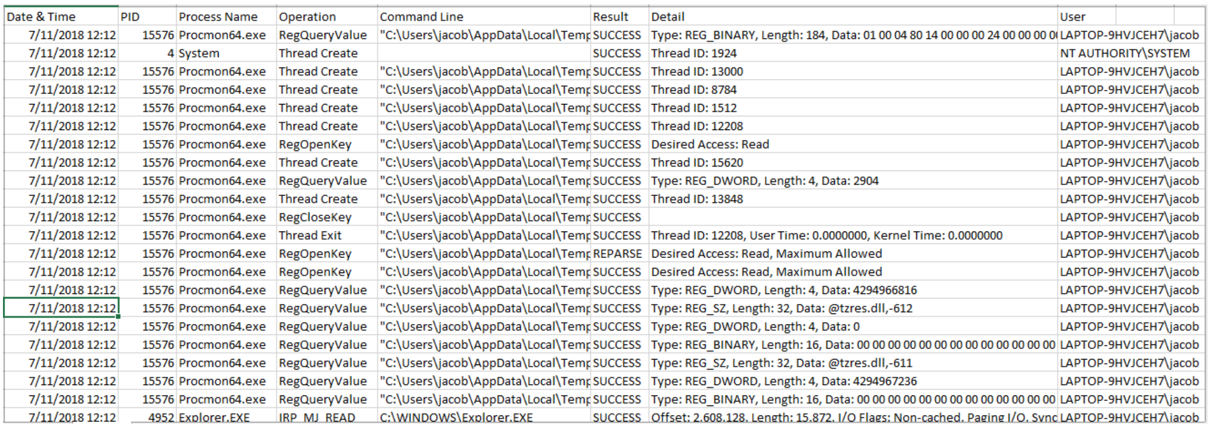
```
C:\Users\ADMIN\Desktop\proj\win32>python tracermon.py 10
Traceback (most recent call last):
  File "tracermon.py", line 9, in <module>
    import psutil
ModuleNotFoundError: No module named 'psutil'

C:\Users\ADMIN\Desktop\proj\win32>
```

Figure 1.4: The error that occurs if the tracing program is run on a Windows system without the psutil Python module installed



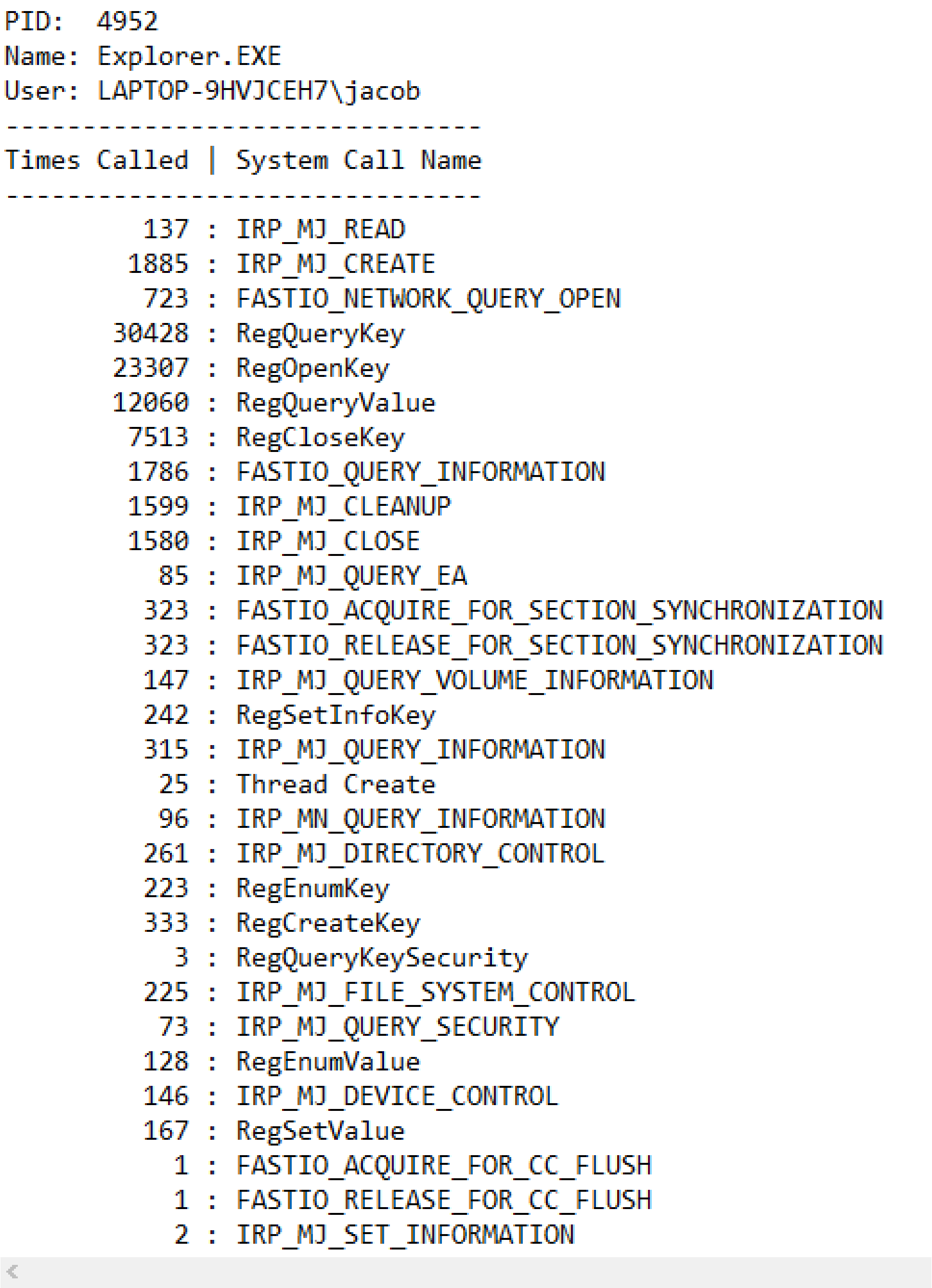
```
tyProject-master>python multiplayer.py
Windows
No custom timeframe set, defaulting to 10 seconds.
Current date and time:  20181107121247
Capturing all calls...

Converting raw output to CSV...
(Depending on time run and programs open, it may take a while)

Sorting statistics...
Writing statistics to file
```

Figure 2.1: Windows runtime output



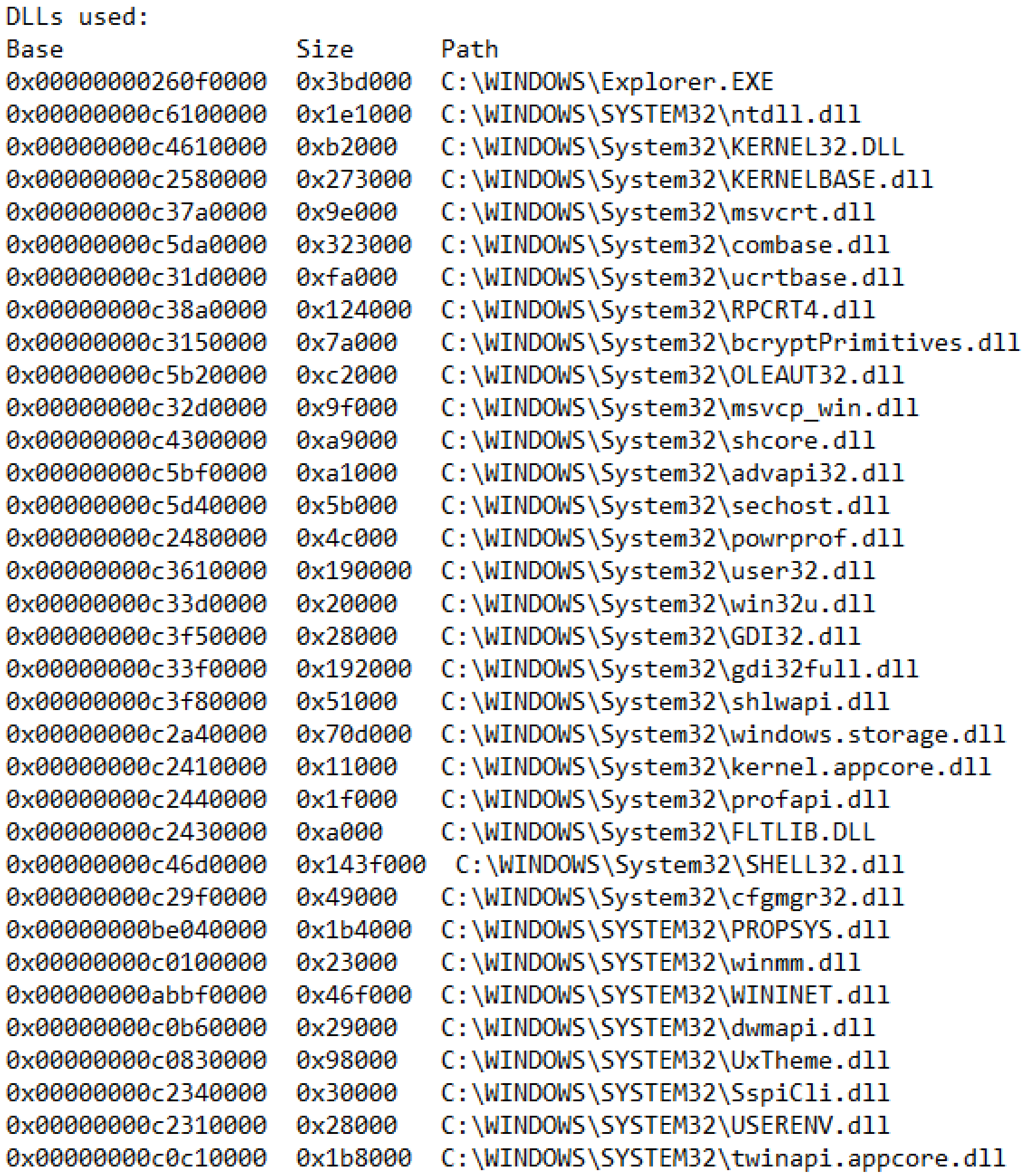
Date & Time	PID	Process Name	Operation	Command Line	Result	Detail	User
7/11/2018 12:12	15576	Procmon64.exe	RegQueryValue	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Type: REG_BINARY, Length: 184, Data: 01 00 04 80 14 00 00 00 24 00 00 00 0	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	4	System	Thread Create		SUCCESS	Thread ID: 1924	NT AUTHORITY\SYSTEM
7/11/2018 12:12	15576	Procmon64.exe	Thread Create	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Thread ID: 13000	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	Thread Create	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Thread ID: 8784	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	Thread Create	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Thread ID: 1512	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	Thread Create	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Thread ID: 12208	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	RegOpenKey	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Desired Access: Read	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	Thread Create	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Thread ID: 15620	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	RegQueryValue	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Type: REG_DWORD, Length: 4, Data: 2904	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	Thread Create	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Thread ID: 13848	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	RegCloseKey	"C:\Users\jacob\AppData\Local\Temp	SUCCESS		LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	Thread Exit	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Thread ID: 12208, User Time: 0.0000000, Kernel Time: 0.0000000	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	RegOpenKey	"C:\Users\jacob\AppData\Local\Temp	REPARSE	Desired Access: Read, Maximum Allowed	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	RegOpenKey	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Desired Access: Read, Maximum Allowed	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	RegQueryValue	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Type: REG_DWORD, Length: 4, Data: 4294966816	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	RegQueryValue	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Type: REG_SZ, Length: 32, Data: @tzres.dll,-612	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	RegQueryValue	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	RegQueryValue	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Type: REG_BINARY, Length: 16, Data: 00 00 00 00 00 00 00 00 00 00 00 00 00	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	RegQueryValue	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Type: REG_SZ, Length: 32, Data: @tzres.dll,-611	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	RegQueryValue	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Type: REG_DWORD, Length: 4, Data: 4294967236	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	15576	Procmon64.exe	RegQueryValue	"C:\Users\jacob\AppData\Local\Temp	SUCCESS	Type: REG_BINARY, Length: 16, Data: 00 00 00 00 00 00 00 00 00 00 00 00 00	LAPTOP-9HVJCEH7\jacob
7/11/2018 12:12	4952	Explorer.EXE	IRP_MJ_READ	C:\WINDOWS\Explorer.EXE	SUCCESS	Offset: 2,608,128, Length: 15,872, I/O Flags: Non-cached, Paging I/O, Sync	LAPTOP-9HVJCEH7\jacob

Figure 2.2: Sample Windows CSV output



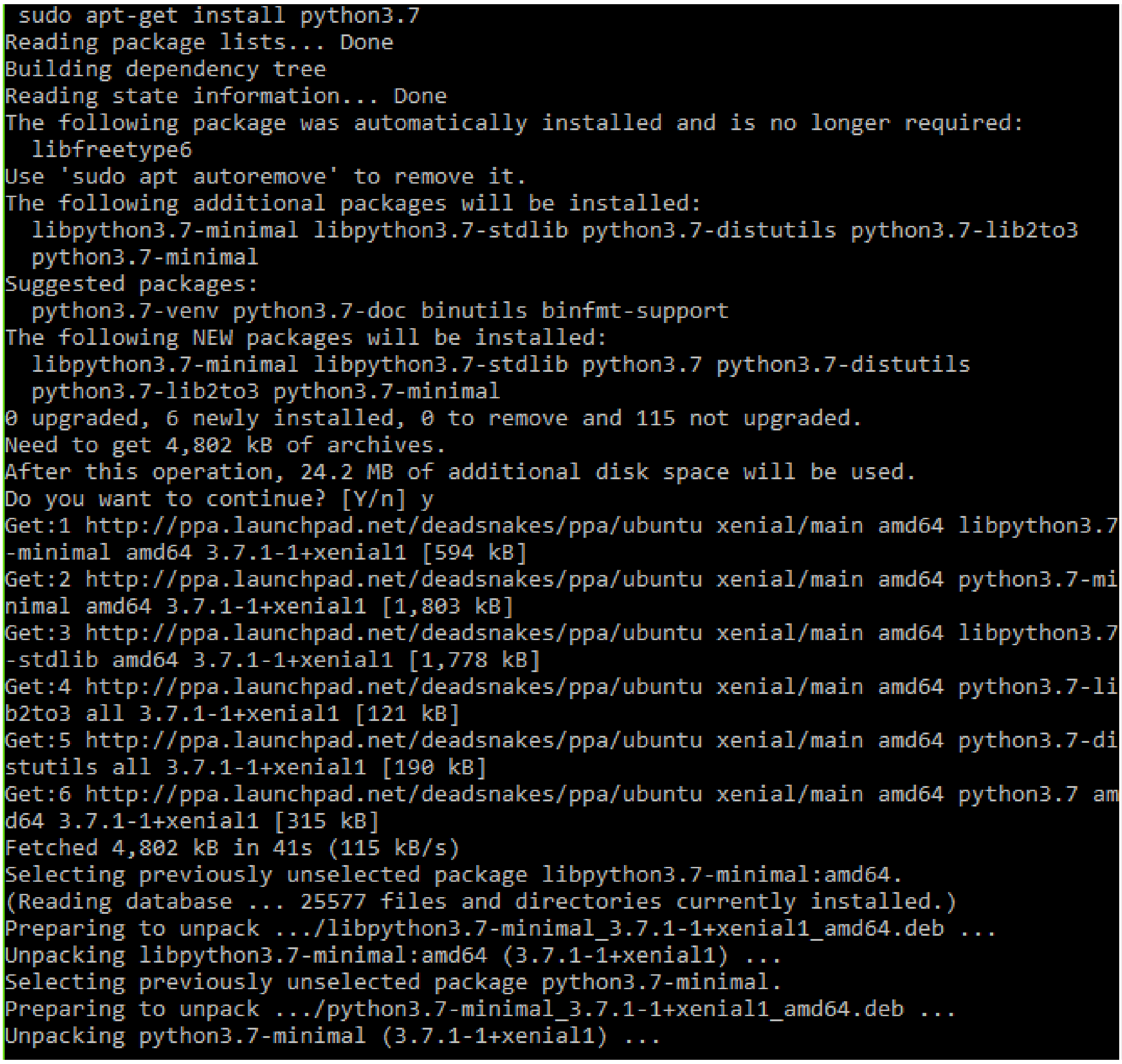
```
PID:  4952
Name: Explorer.EXE
User: LAPTOP-9HVJCEH7\jacob
-------------------------------
Times Called | System Call Name
-------------------------------
         137 : IRP_MJ_READ
        1885 : IRP_MJ_CREATE
         723 : FASTIO_NETWORK_QUERY_OPEN
       30428 : RegQueryKey
       23307 : RegOpenKey
       12060 : RegQueryValue
        7513 : RegCloseKey
        1786 : FASTIO_QUERY_INFORMATION
        1599 : IRP_MJ_CLEANUP
        1580 : IRP_MJ_CLOSE
          85 : IRP_MJ_QUERY_EA
         323 : FASTIO_ACQUIRE_FOR_SECTION_SYNCHRONIZATION
         323 : FASTIO_RELEASE_FOR_SECTION_SYNCHRONIZATION
         147 : IRP_MJ_QUERY_VOLUME_INFORMATION
         242 : RegSetInfoKey
         315 : IRP_MJ_QUERY_INFORMATION
          25 : Thread Create
          96 : IRP_MN_QUERY_INFORMATION
         261 : IRP_MJ_DIRECTORY_CONTROL
         223 : RegEnumKey
         333 : RegCreateKey
           3 : RegQueryKeySecurity
         225 : IRP_MJ_FILE_SYSTEM_CONTROL
          73 : IRP_MJ_QUERY_SECURITY
         128 : RegEnumValue
         146 : IRP_MJ_DEVICE_CONTROL
         167 : RegSetValue
           1 : FASTIO_ACQUIRE_FOR_CC_FLUSH
           1 : FASTIO_RELEASE_FOR_CC_FLUSH
           2 : IRP_MJ_SET_INFORMATION
```

Figure 2.3: Windows system call statistics



```
DLLs used:
Base                Size      Path
0x00000000260f0000  0x3bd000  C:\WINDOWS\Explorer.EXE
0x00000000c6100000  0x1e1000  C:\WINDOWS\SYSTEM32\ntdll.dll
0x00000000c4610000  0xb2000   C:\WINDOWS\System32\KERNEL32.DLL
0x00000000c2580000  0x273000  C:\WINDOWS\System32\KERNELBASE.dll
0x00000000c37a0000  0x9e000   C:\WINDOWS\System32\msvcrt.dll
0x00000000c5da0000  0x323000  C:\WINDOWS\System32\combase.dll
0x00000000c31d0000  0xfa000   C:\WINDOWS\System32\ucrtbase.dll
0x00000000c38a0000  0x124000  C:\WINDOWS\System32\RPCRT4.dll
0x00000000c3150000  0x7a000   C:\WINDOWS\System32\bcryptPrimitives.dll
0x00000000c5b20000  0xc2000   C:\WINDOWS\System32\OLEAUT32.dll
0x00000000c32d0000  0x9f000   C:\WINDOWS\System32\msvcp_win.dll
0x00000000c4300000  0xa9000   C:\WINDOWS\System32\shcore.dll
0x00000000c5bf0000  0xa1000   C:\WINDOWS\System32\advapi32.dll
0x00000000c5d40000  0x5b000   C:\WINDOWS\System32\sechost.dll
0x00000000c2480000  0x4c000   C:\WINDOWS\System32\powrprof.dll
0x00000000c3610000  0x190000  C:\WINDOWS\System32\user32.dll
0x00000000c33d0000  0x20000   C:\WINDOWS\System32\win32u.dll
0x00000000c3f50000  0x28000   C:\WINDOWS\System32\GDI32.dll
0x00000000c33f0000  0x192000  C:\WINDOWS\System32\gdi32full.dll
0x00000000c3f80000  0x51000   C:\WINDOWS\System32\shlwapi.dll
0x00000000c2a40000  0x70d000  C:\WINDOWS\System32\windows.storage.dll
0x00000000c2410000  0x11000   C:\WINDOWS\System32\kernel.appcore.dll
0x00000000c2440000  0x1f000   C:\WINDOWS\System32\profapi.dll
0x00000000c2430000  0xa000    C:\WINDOWS\System32\FLTLIB.DLL
0x00000000c46d0000  0x143f000  C:\WINDOWS\System32\SHELL32.dll
0x00000000c29f0000  0x49000   C:\WINDOWS\System32\cfgmgr32.dll
0x00000000be040000  0x1b4000  C:\WINDOWS\SYSTEM32\PROPSYS.dll
0x00000000c0100000  0x23000   C:\WINDOWS\SYSTEM32\winmm.dll
0x00000000abbf0000  0x46f000  C:\WINDOWS\SYSTEM32\WININET.dll
0x00000000c0b60000  0x29000   C:\WINDOWS\SYSTEM32\dwmapi.dll
0x00000000c0830000  0x98000   C:\WINDOWS\SYSTEM32\UxTheme.dll
0x00000000c2340000  0x30000   C:\WINDOWS\SYSTEM32\SspiCli.dll
0x00000000c2310000  0x28000   C:\WINDOWS\SYSTEM32\USERENV.dll
0x00000000c0c10000  0x1b8000  C:\WINDOWS\SYSTEM32\twinapi.appcore.dll
```

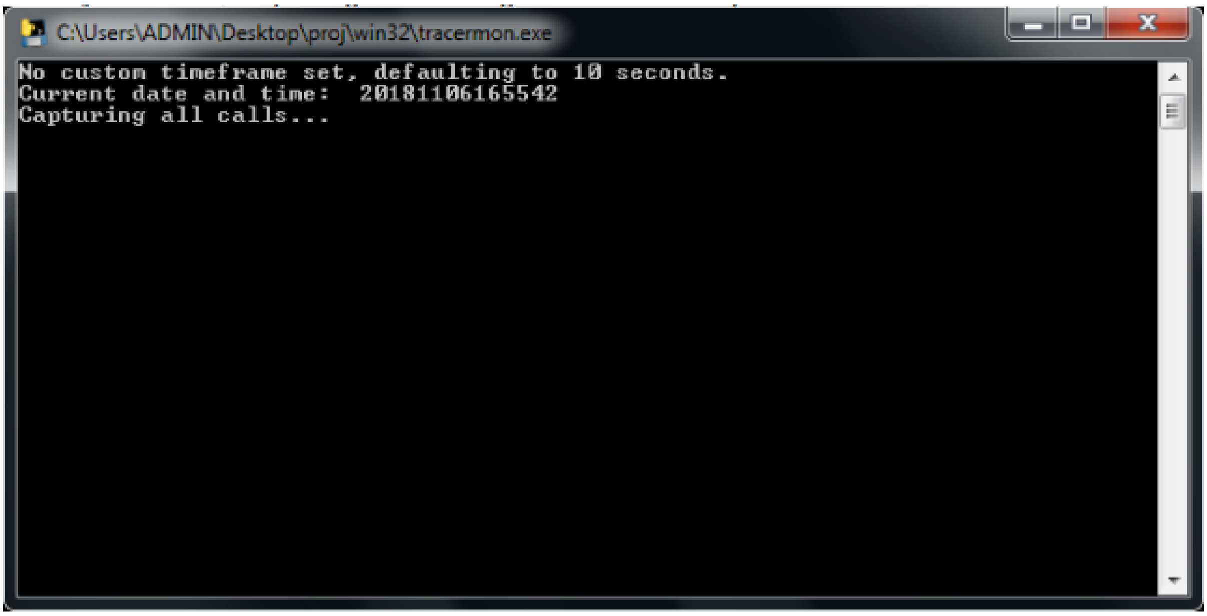
Figure 2.4: List of DLLs used by a process



```
 sudo apt-get install python3.7
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfreetype6
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libpython3.7-minimal libpython3.7-stdlib python3.7-distutils python3.7-lib2to3
  python3.7-minimal
Suggested packages:
  python3.7-venv python3.7-doc binutils binfmt-support
The following NEW packages will be installed:
  libpython3.7-minimal libpython3.7-stdlib python3.7 python3.7-distutils
  python3.7-lib2to3 python3.7-minimal
0 upgraded, 6 newly installed, 0 to remove and 115 not upgraded.
Need to get 4,802 kB of archives.
After this operation, 24.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu xenial/main amd64 libpython3.7
-minimal amd64 3.7.1-1+xenial1 [594 kB]
Get:2 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu xenial/main amd64 python3.7-mi
nimal amd64 3.7.1-1+xenial1 [1,803 kB]
Get:3 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu xenial/main amd64 libpython3.7
-stdlib amd64 3.7.1-1+xenial1 [1,778 kB]
Get:4 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu xenial/main amd64 python3.7-li
b2to3 all 3.7.1-1+xenial1 [121 kB]
Get:5 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu xenial/main amd64 python3.7-di
stutils all 3.7.1-1+xenial1 [190 kB]
Get:6 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu xenial/main amd64 python3.7 am
d64 3.7.1-1+xenial1 [315 kB]
Fetched 4,802 kB in 41s (115 kB/s)
Selecting previously unselected package libpython3.7-minimal:amd64.
(Reading database ... 25577 files and directories currently installed.)
Preparing to unpack .../libpython3.7-minimal_3.7.1-1+xenial1_amd64.deb ...
Unpacking libpython3.7-minimal:amd64 (3.7.1-1+xenial1) ...
Selecting previously unselected package python3.7-minimal.
Preparing to unpack .../python3.7-minimal_3.7.1-1+xenial1_amd64.deb ...
Unpacking python3.7-minimal (3.7.1-1+xenial1) ...
```

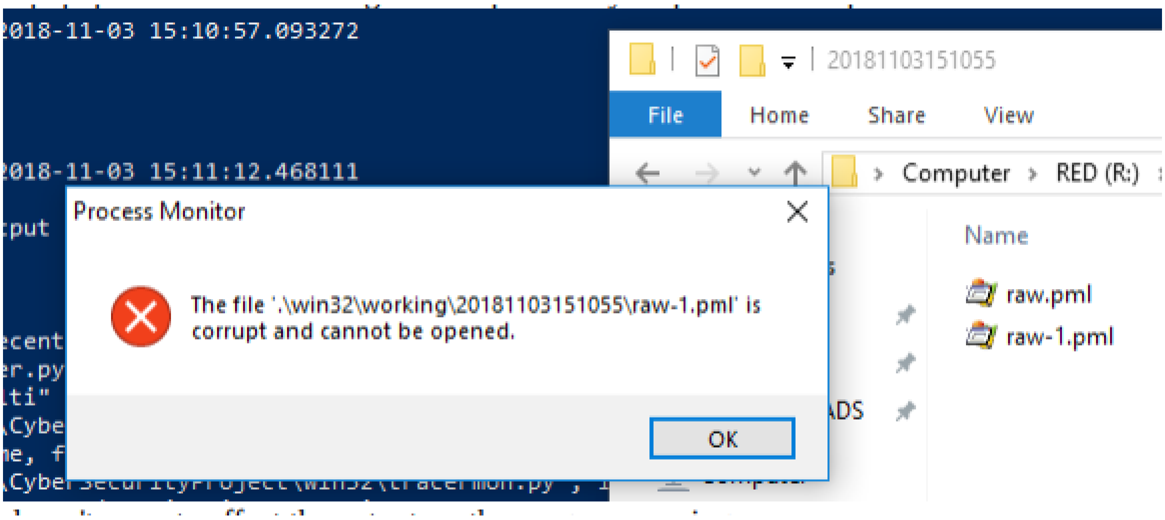
Figure 3.1: Installation of Python 3.7 on Ubuntu

Figure 4.1: Sample output of the Linux operation



```
No custom timeframe set, defaulting to 10 seconds.
Current date and time:  20181106165542
Capturing all calls...
```

Figure 5.1: The Windows program may hang while attempting to run the ListDLLs process

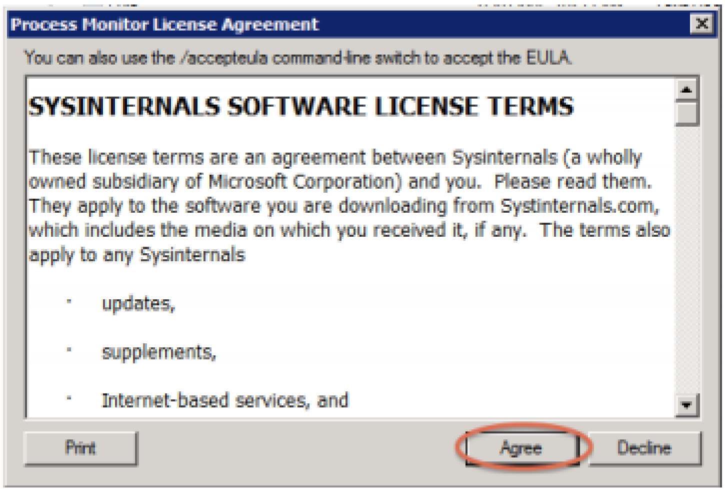


```
2018-11-03 15:10:57.093272

2018-11-03 15:11:12.468111
```

Process Monitor

The file '.\win32\working\20181103151055\raw-1.pml' is corrupt and cannot be opened.

Figure 5.2: Occasionally the system warns part of the output is corrupted. However, this does not seem to have any actual affect



Process Monitor License Agreement

You can also use the /accepteula command-line switch to accept the EULA.

SYSINTERNALS SOFTWARE LICENSE TERMS

These license terms are an agreement between Sysinternals (a wholly owned subsidiary of Microsoft Corporation) and you. Please read them. They apply to the software you are downloading from Systinternals.com, which includes the media on which you received it, if any. The terms also apply to any Sysinternals

- updates,
- supplements,
- Internet-based services, and

Figure 5.3: Despite the use of the “/accepteula” switch this popup may still occasionally appear on the Windows distribution