PyREBox

Information Security Inc.

Contents

- About PyREBox
- Building PyREBox
- Usage
- FAQ
- References

About PyREBox

- PyREBox is a Python scriptable Reverse Engineering sandbox. It is based on QEMU, and its goal is to aid reverse engineering by providing dynamic analysis and debugging capabilities from a different perspective. PyREBox allows to inspect a running QEMU VM, modify its memory or registers, and to instrument its execution, by creating simple scripts in python to automate any kind of analysis.

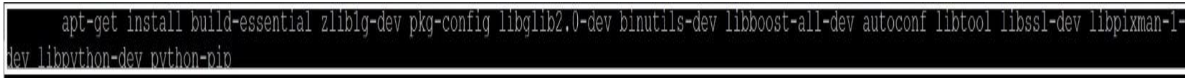
Building PyREBox

◎ Testing environment

Host OS:Ubuntu 16.04

Guest OS: Windows 7 32 and 64Bit

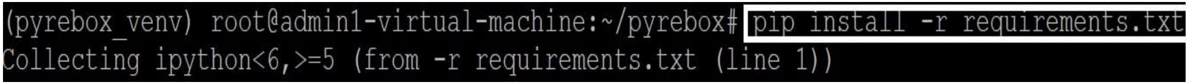
◎ Installing dependencies for Debian based distributions



```
     apt-get install build-essential zlib1g-dev pkg-config libglib2.0-dev binutils-dev libboost-all-dev autoconf libtool libssl-dev libpixman-1-
dev libpython-dev python-pip
```

◎ Required python packages

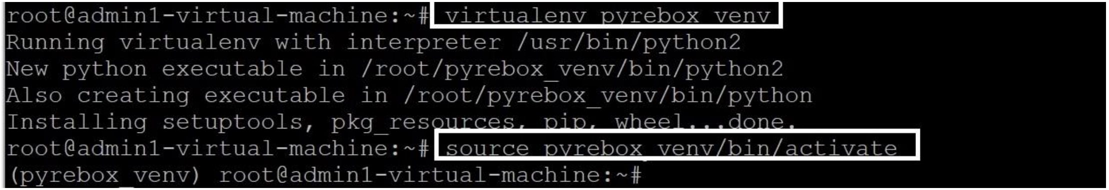
ipython>=5,<6 sphinx sphinx-autobuild prettytable pefile capstone distorm3 pycrypto pytz



```
(pyrebox_venv) root@admin1-virtual-machine:~/pyrebox# pip install -r requirements.txt
Collecting ipython<6,>=5 (from -r requirements.txt (line 1))
```

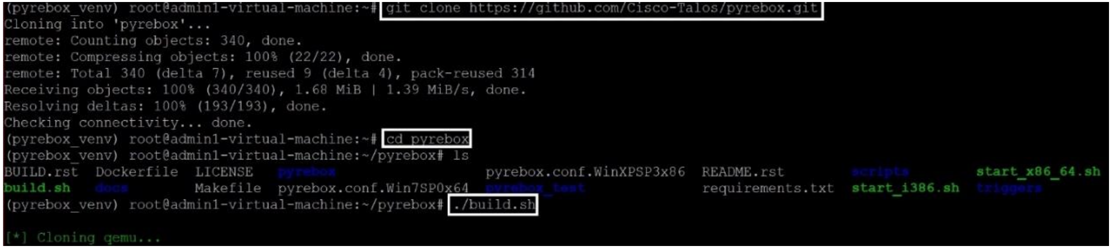
Building PyREBox

◎ Create a virtual environment for PyREBox

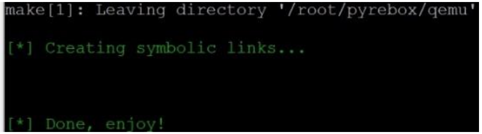


```
root@admin1-virtual-machine:~# virtualenv pyrebox_venv
Running virtualenv with interpreter /usr/bin/python2
New python executable in /root/pyrebox_venv/bin/python2
Also creating executable in /root/pyrebox_venv/bin/python
Installing setuptools, pkg_resources, pip, wheel...done.
root@admin1-virtual-machine:~# source pyrebox_venv/bin/activate
(pyrebox_venv) root@admin1-virtual-machine:~#
```

◎ Download and install pyrebox



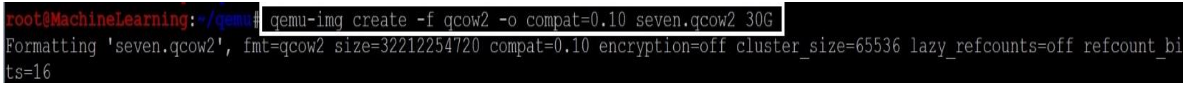
```
(pyrebox_venv) root@admin1-virtual-machine:~# git clone https://github.com/Cisco-Talos/pyrebox.git
Cloning into 'pyrebox'...
remote: Counting objects: 340, done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 340 (delta 7), reused 9 (delta 4), pack-reused 314
Receiving objects: 100% (340/340), 1.68 MiB | 1.39 MiB/s, done.
Resolving deltas: 100% (193/193), done.
Checking connectivity... done.
(pyrebox_venv) root@admin1-virtual-machine:~# cd pyrebox
(pyrebox_venv) root@admin1-virtual-machine:~/pyrebox# ls
BUILD.rst  Dockerfile  LICENSE   pyrebox                  pyrebox.conf.WinXPSP3x86  README.rst        scripts        start_x86_64.sh
build.sh   docs        Makefile  pyrebox.conf.Win7SP0x64  pyrebox_test              requirements.txt  start_i386.sh  triggers
(pyrebox_venv) root@admin1-virtual-machine:~/pyrebox# ./build.sh

[*] Cloning qemu...
```



```
make[1]: Leaving directory '/root/pyrebox/qemu'

[*] Creating symbolic links...


[*] Done, enjoy!
```

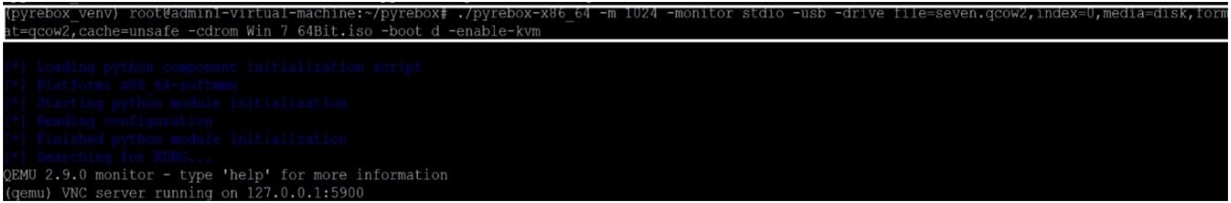
Usage

- Create a VM image for PyREBox

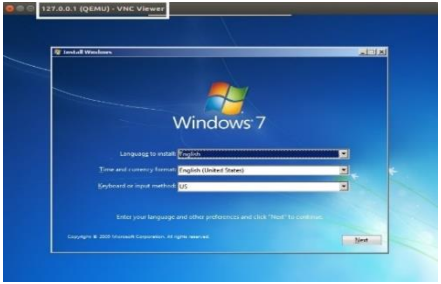


```
root@MachineLearning:~/qemu# qemu-img create -f qcow2 -o compat=0.10 seven.qcow2 30G
Formatting 'seven.qcow2', fmt=qcow2 size=32212254720 compat=0.10 encryption=off cluster_size=65536 lazy_refcounts=off refcount_bi
ts=16
```

- Install guest OS



```
(pyrebox_venv) root@admin1-virtual-machine:~/pyrebox# ./pyrebox-x86_64 -m 1024 -monitor stdio -usb -drive file=seven.qcow2,index=0,media=disk,form
at=qcow2,cache=unsafe -cdrom Win 7 64Bit.iso -boot d -enable-kvm

[*] Loading python component initialization script
[*] Platform: x86_64-softmmu
[*] Starting python module initialization
[*] Reading configuration
[*] Finished python module initialization
[*] Searching for KDBG...
QEMU 2.9.0 monitor - type 'help' for more information
(qemu) VNC server running on 127.0.0.1:5900
```



Language to install: English

Time and currency format: English (United States)

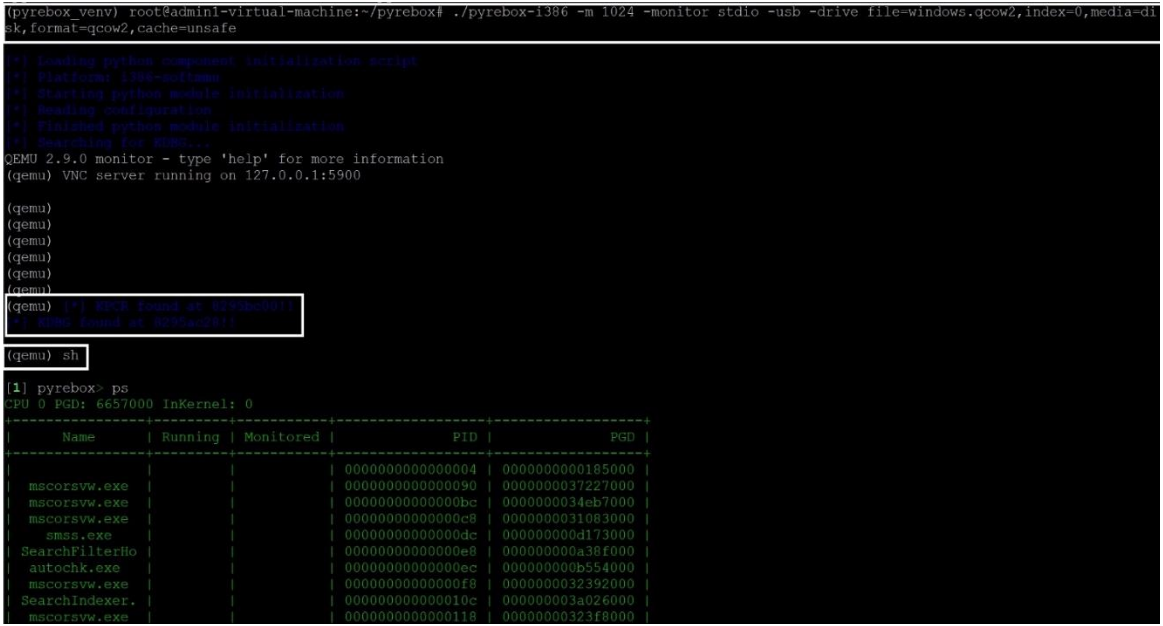
Keyboard or input method: US

Enter your language and other preferences and click "Next" to continue.

Copyright © 2009 Microsoft Corporation. All rights reserved.

Usage

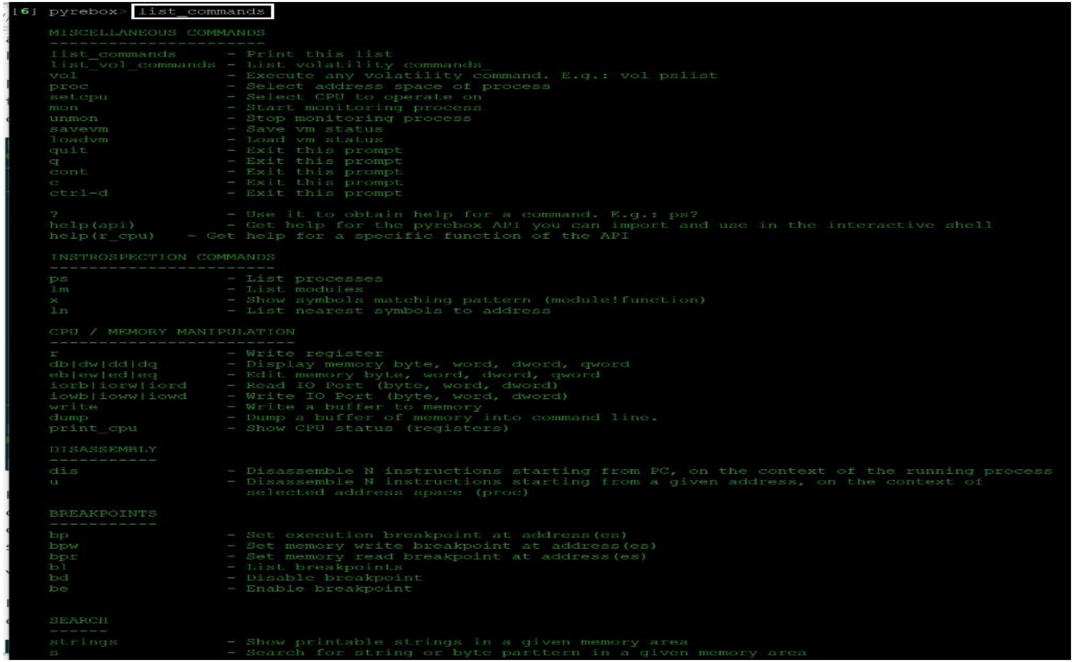
- Start the PyREBox shell



```
(pyrebox_venv) root@admin1-virtual-machine:~/pyrebox# ./pyrebox-i386 -m 1024 -monitor stdio -usb -drive file=windows.qcow2,index=0,media=di
sk,format=qcow2,cache=unsafe

[*] Loading python component initialization script
[*] Platform: i386-softmmu
[*] Starting python module initialization
[*] Reading configuration
[*] Finished python module initialization
[*] Searching for KDBG...
QEMU 2.9.0 monitor - type 'help' for more information
(qemu) VNC server running on 127.0.0.1:5900

(qemu)
(qemu)
(qemu)
(qemu)
(qemu)
(qemu)
(qemu) [*] KPCR found at 8295bc00!!
[*] KDBG found at 8295ac28!!

(qemu) sh

[1] pyrebox> ps
CPU 0 PGD: 6657000 InKernel: 0
+-----------------+---------+-----------+------------------+------------------+
|      Name       | Running | Monitored |              PID |              PGD |
+-----------------+---------+-----------+------------------+------------------+
|                 |         |           | 0000000000000004 | 0000000000185000 |
|   mscorsvw.exe  |         |           | 0000000000000090 | 0000000037227000 |
|   mscorsvw.exe  |         |           | 00000000000000bc | 0000000034eb7000 |
|   mscorsvw.exe  |         |           | 00000000000000c8 | 0000000031083000 |
|     smss.exe    |         |           | 00000000000000dc | 000000000d173000 |
| SearchFilterHo  |         |           | 00000000000000e8 | 000000000a38f000 |
|   autochk.exe   |         |           | 00000000000000ec | 000000000b554000 |
|   mscorsvw.exe  |         |           | 00000000000000f8 | 0000000032392000 |
| SearchIndexer.  |         |           | 000000000000010c | 000000003a026000 |
|   mscorsvw.exe  |         |           | 0000000000000118 | 00000000323f8000 |
```

Usage

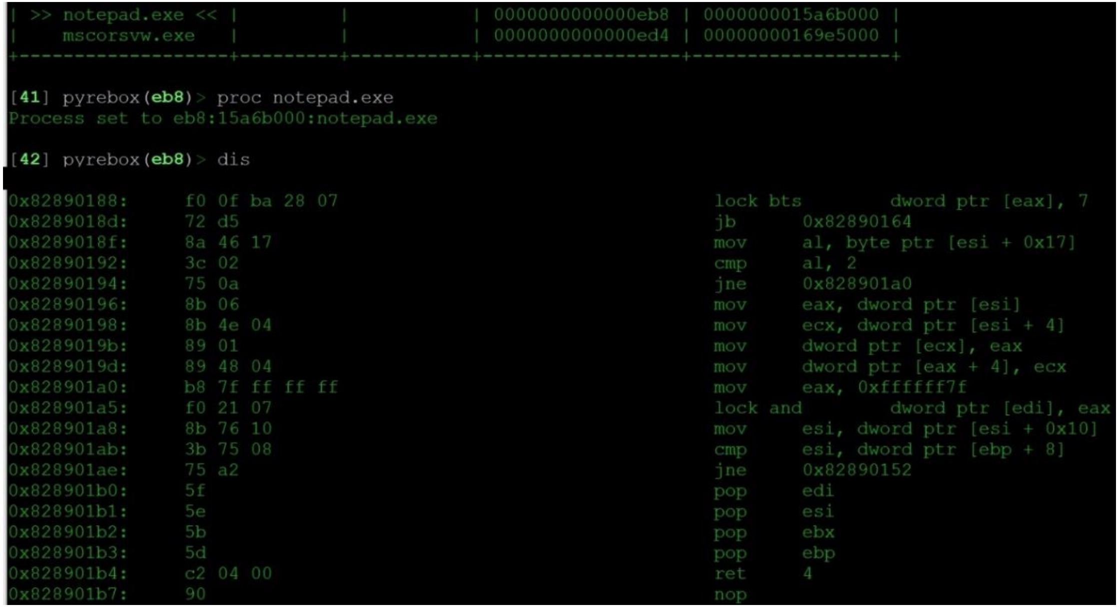
- ◎ List commands



```
[6] pyrebox> list_commands
    MISCELLANEOUS COMMANDS
    -----------------------
    list_commands       - Print this list
    list_vol_commands - List volatility commands_
    vol                 - Execute any volatility command. E.g.: vol pslist
    proc                - Select address space of process
    setcpu              - Select CPU to operate on
    mon                 - Start monitoring process
    unmon               - Stop monitoring process
    savevm              - Save vm status
    loadvm              - Load vm status
    quit                - Exit this prompt
    q                   - Exit this prompt
    cont                - Exit this prompt
    c                   - Exit this prompt
    ctrl-d              - Exit this prompt

    ?                   - Use it to obtain help for a command. E.g.: ps?
    help(api)           - Get help for the pyrebox API you can import and use in the interactive shell
    help(r_cpu)    - Get help for a specific function of the API
    INSTROSPECTION COMMANDS
    -----------------------
    ps                  - List processes
    lm                  - List modules
    x                   - Show symbols matching pattern (module!function)
    ln                  - List nearest symbols to address

    CPU / MEMORY MANIPULATION
    -------------------------
    r                   - Write register
    db|dw|dd|dq         - Display memory byte, word, dword, qword
    eb|ew|ed|eq         - Edit memory byte, word, dword, qword
    iorb|iorw|iord      - Read IO Port (byte, word, dword)
    iowb|ioww|iowd      - Write IO Port (byte, word, dword)
    write               - Write a buffer to memory
    dump                - Dump a buffer of memory into command line.
    print_cpu           - Show CPU status (registers)

    DISASSEMBLY
    -----------
    dis                 - Disassemble N instructions starting from PC, on the context of the running process
    u                   - Disassemble N instructions starting from a given address, on the context of
                          selected address space (proc)

    BREAKPOINTS
    -----------
    bp                  - Set execution breakpoint at address(es)
    bpw                 - Set memory write breakpoint at address(es)
    bpr                 - Set memory read breakpoint at address(es)
    bl                  - List breakpoints
    bd                  - Disable breakpoint
    be                  - Enable breakpoint


    SEARCH
    ------
    strings             - Show printable strings in a given memory area
    s                   - Search for string or byte parttern in a given memory area
```

Usage

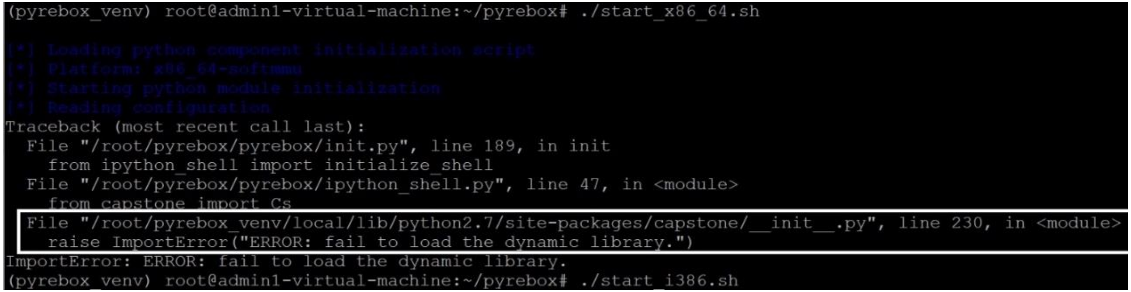
- Examine a process



```
| >> notepad.exe << |         |           | 0000000000000eb8 | 0000000015a6b000 |
|    mscorsvw.exe   |         |           | 0000000000000ed4 | 00000000169e5000 |
+-------------------+---------+-----------+------------------+------------------+

[41] pyrebox(eb8)> proc notepad.exe
Process set to eb8:15a6b000:notepad.exe

[42] pyrebox(eb8)> dis

0x82890188:     f0 0f ba 28 07                                  lock bts        dword ptr [eax], 7
0x8289018d:     72 d5                                           jb      0x82890164
0x8289018f:     8a 46 17                                        mov     al, byte ptr [esi + 0x17]
0x82890192:     3c 02                                           cmp     al, 2
0x82890194:     75 0a                                           jne     0x828901a0
0x82890196:     8b 06                                           mov     eax, dword ptr [esi]
0x82890198:     8b 4e 04                                        mov     ecx, dword ptr [esi + 4]
0x8289019b:     89 01                                           mov     dword ptr [ecx], eax
0x8289019d:     89 48 04                                        mov     dword ptr [eax + 4], ecx
0x828901a0:     b8 7f ff ff ff                                  mov     eax, 0xffffff7f
0x828901a5:     f0 21 07                                        lock and        dword ptr [edi], eax
0x828901a8:     8b 76 10                                        mov     esi, dword ptr [esi + 0x10]
0x828901ab:     3b 75 08                                        cmp     esi, dword ptr [ebp + 8]
0x828901ae:     75 a2                                           jne     0x82890152
0x828901b0:     5f                                              pop     edi
0x828901b1:     5e                                              pop     esi
0x828901b2:     5b                                              pop     ebx
0x828901b3:     5d                                              pop     ebp
0x828901b4:     c2 04 00                                        ret     4
0x828901b7:     90                                              nop
```

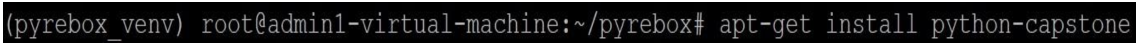
FAQ

◎ If getting the following error



```
(pyrebox_venv) root@admin1-virtual-machine:~/pyrebox# ./start_x86_64.sh

[*] Loading python component initialization script
[*] Platform: x86_64-softmmu
[*] Starting python module initialization
[*] Reading configuration
Traceback (most recent call last):
  File "/root/pyrebox/pyrebox/init.py", line 189, in init
    from ipython_shell import initialize_shell
  File "/root/pyrebox/pyrebox/ipython_shell.py", line 47, in <module>
    from capstone import Cs
  File "/root/pyrebox_venv/local/lib/python2.7/site-packages/capstone/__init__.py", line 230, in <module>
    raise ImportError("ERROR: fail to load the dynamic library.")
ImportError: ERROR: fail to load the dynamic library.
(pyrebox_venv) root@admin1-virtual-machine:~/pyrebox# ./start_i386.sh
```

To resolve it > Install capstone with apt



```
(pyrebox_venv) root@admin1-virtual-machine:~/pyrebox# apt-get install python-capstone
```

References

- Github
- https://github.com/Cisco-Talos/pyrebox