DRAKVUF

Information Security Inc.

iSEC
information security inc.

Contents

- About DRAKVUF
- Why DRAKVUF?
- Hardware requirements
- Guest OS compatibility
- Testing environment
- DRAKVUF Installation
- DRAKVUF Demo
- References

About DRAKVUF

- DRAKVUF is a virtualization based agentless black-box binary analysis system. DRAKVUF allows for in-depth execution tracing of arbitrary binaries (including operating systems), all without having to install any special software within the virtual machine used for analysis

Why DRAKVUF?

- DRAKVUF provides a perfect platform for stealthy malware analysis as its footprint is nearly undetectable from the malware's perspective

Hardware requirements

- DRAKVUF uses hardware virtualization extensions found in Intel CPUs. You will need an Intel CPU with virtualization support (VT-x) and with Extended Page Tables (EPT). DRAKVUF is not going to work on any other CPUs (such as AMD) or on Intel CPUs without the required virtualization extensions

Guest OS compatibility

- DRAKVUF currently supports:

  - Windows 7 - 8, both 32 and 64-bit
  - Windows 10 64-bit
  - Linux 2.6.x - 4.x, both 32-bit and 64-bit

Testing environment

- Linux Ubuntu 5.4.0-6ubuntu1~16.04.4
- CPU: Intel Xeon E5-1620 with virtualization support (VT-x) and with Extended Page Tables (EPT)
- Xen hypervisor 4.9

DRAKVUF Installation

▲ Install the required packages

```
malw@ubuntu:~$ sudo apt-get install wget git bcc bin86 gawk bridge-utils iproute libcurl3 libcurl4-openssl-dev bzip2 pciutils-dev build-essential make gcc cla
ng libc6-dev libc6-dev-i386 linux-libc-dev zlib1g-dev python-dev python-pip libncurses5-dev patch libvncserver-dev libssl-dev libsdl-dev iasl libbz2-dev e2fsl
ibs-dev git-core uuid-dev ocaml libx11-dev bison flex ocaml-findlib xz-utils gettext libyajl-dev libpixman-1-dev libaio-dev libfdt-dev cabextract libglib2.0-d
ev autoconf automake libtool check libjson-c-dev libfuse-dev checkpolicy liblzma-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

▲ Download drakvuf, rekall, Xen

```
git clone https://github.com/tklengyel/drakvuf
cd drakvuf/
git submodule init
git submodule update
```

▲ Set up Xen

```
cd xen
./configure --enable-githttp
make -j4 dist-xen
make -j4 dist-tools
```

DRAKVUF Installation

▲ Install Xen

$sudo su

```
make -j4 install-xen
make -j4 install-tools
echo "GRUB_CMDLINE_XEN_DEFAULT=\"dom0_mem=4096M,max:4096M dom0_max_vcpus=2 dom0_vcpus_pin=
etc/default/grub
echo "/usr/local/lib" > /etc/ld.so.conf.d/xen.conf
ldconfig
echo "none /proc/xen xenfs defaults,nofail 0 0" >> /etc/fstab
echo "xen-evtchn" >> /etc/modules
echo "xen-privcmd" >> /etc/modules
update-rc.d xencommons defaults 19 18
update-rc.d xendomains defaults 21 20
update-rc.d xen-watchdog defaults 22 23
```

▲ Make xen boot before the kernel

◎ cd /etc/grub.d/;mv 20_linux_xen 09_linux_xen

▲ Finalize the setup

◎ update-grub

◎ reboot

DRAKVUF Installation

▲ Verify Xen installation

```
root@ubuntu:/home/adi# xen-detect
Running in PV context on Xen v4.9.
root@ubuntu:/home/adi# xl list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0  4096     2     r-----     526.7
root@ubuntu:/home/adi#
```

▲ Setup an LVM volume Group to hold the VMs, then create a volume

```
root@ubuntu:/home/adi# lvcreate -L30G -n windows10-64 vg
  Logical volume "windows10-64" created.
```

DRAKVUF Installation

▲ Configure Xen bridge interface (xenbr0)

```
root@XeonPowerful:~/drakvuf# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eno1
iface eno1 inet manual
        #address 192.168.86.8
        #netmask 255.255.255.0
        #network 192.168.86.0
        #broadcast 192.168.86.255
        #gateway 192.168.86.86
        # dns-* options are implemented by the resolvconf package, if installed
        #dns-nameservers 8.8.8.8

auto xenbr0
iface xenbr0 inet static
  bridge_ports eno1
  address 192.168.86.8
  broadcast 192.168.86.255
  netmask 255.255.255.0
  network 192.168.86.0
  gateway 192.168.86.86
  dns-nameservers 8.8.8.8
```

DRAKVUF Installation

▲ Install Windows from ISO

Create vm config file and create the virtual machine

```
root@XeonPowerful:~# xl create Windows7.hvm
```

```
root@XeonPowerful:~# cat Windows7.hvm
arch = 'x86_64'
name = "windowsseven"
seclabel='drakvuf:vm_r:drakvuf_domU_t'
maxmem = 3000
memory = 3000
vcpus = 2
maxcpus = 2
builder = "hvm"
boot = "cd"
hap = 1
acpi = 1
on_poweroff = "destroy"
on_reboot = "destroy"
on_crash = "destroy"
vnc=1
vnclisten="0.0.0.0"
usb = 1
usbdevice = "tablet"
altp2m = 2
shadow_memory = 16
audio=1
soundhw='hda'
vif = [ 'type=ioemu,model=e1000,bridge=xenbr0,mac=10:60:4b:7e:ed:f7,ip=192.168.86.80' ]
disk = [ 'phy:/dev/GROUP1/LogicVolume,hda,w', 'file:/root/Windowsseven.iso,hdc:cdrom,r' ]
```

▲ Enter the LibVMI folder in the drakvuf folder and build it

```
cd drakvuf/
ls
cd libvmi/
ls
./autogen.sh
./configure --disable-kvm
```

▲ Build and install LibVMI

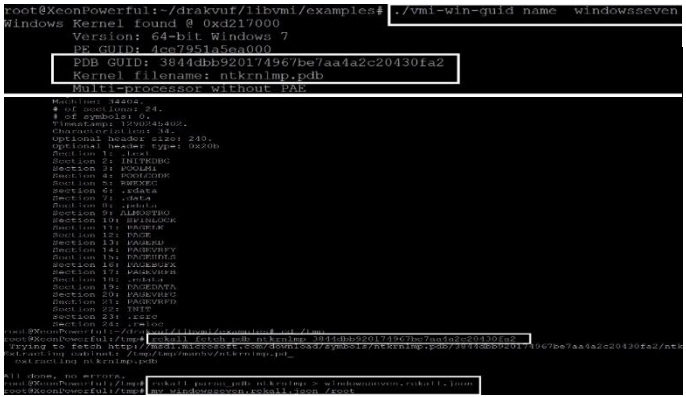
```
sudo make
sudo make install
sudo echo "export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/usr/local/lib" >> ~/.bashrc
```

DRAKVUF Installation

▲ Build and install Rekall

```
cd drakvuf/
cd rekall/
cd rekall-core/
pip install setuptools
pip install --upgrade pip
pip install setuptools
pwd
python setup.py build
python setup.py install
```

▲ Create the Rekall profile for the Windows domain



DRAKVUF Installation

▲ Create the LibVMI config and test it by running vmi-process-list

```
root@XeonPowerful:~/drakvuf/libvmi/examples# cat /etc/libvmi.conf
windowsseven {
    ostype = "Windows";
    rekall_profile = "/root/windowsseven.rekall.json";
}
root@XeonPowerful:~/drakvuf/libvmi/examples# ./vmi-process-list windowsseven
Process listing for VM windowsseven (id=3)
[    4] System (struct addr:fffffa8002392450)
[  180] smss.exe (struct addr:fffffa8002743220)
[  240] csrss.exe (struct addr:fffffa80028056a0)
[  276] wininit.exe (struct addr:fffffa800239aad0)
[  284] csrss.exe (struct addr:fffffa80028424e0)
[  320] services.exe (struct addr:fffffa800287b060)
[  332] lsass.exe (struct addr:fffffa8002aa5060)
[  340] lsm.exe (struct addr:fffffa8002aa6ab0)
[  468] svchost.exe (struct addr:fffffa8002b8e470)
[  524] svchost.exe (struct addr:fffffa8002bdd210)
[  580] svchost.exe (struct addr:fffffa8002bf6940)
[  656] winlogon.exe (struct addr:fffffa8002be7060)
[  696] winpeshl.exe (struct addr:fffffa8002bdf590)
[  724] setup.exe (struct addr:fffffa8002c60760)
```

DRAKVUF Installation

▲ build and install DRAKVUF

```
cd drakvuf/


autoreconf -vi
./configure
make
```

▲ See all options

```
root@XeonPowerful:~/drakvuf# ./src/drakvuf
DRAKVUF v0.5-bb602f6
Required input:
        -r <rekall profile>       The Rekall profile of the OS kernel
        -d <domain ID or name>    The domain's ID or name
Optional inputs:
        -i <injection pid>        The PID of the process to hijack for injection
        -I <injection thread>     The ThreadID in the process to hijack for injection (requires -i)
        -e <inject_exe>           The executable to start with injection
        -t <timeout>              Timeout (in seconds)
        -o <format>               Output format (default or csv)
        -x <plugin>               Don't activate the specified plugin
        -p                        Leave domain paused after DRAKVUF exits
        -w <process name>         Wait with plugin start until process name is detected
        -D <file dump folder>     Folder where extracted files should be stored at
        -T <rekall profile>       The Rekall profile for tcpip.sys
        -s                        Hide Hypervisor bits and signature in CPUID
```

DRAKVUF Demo

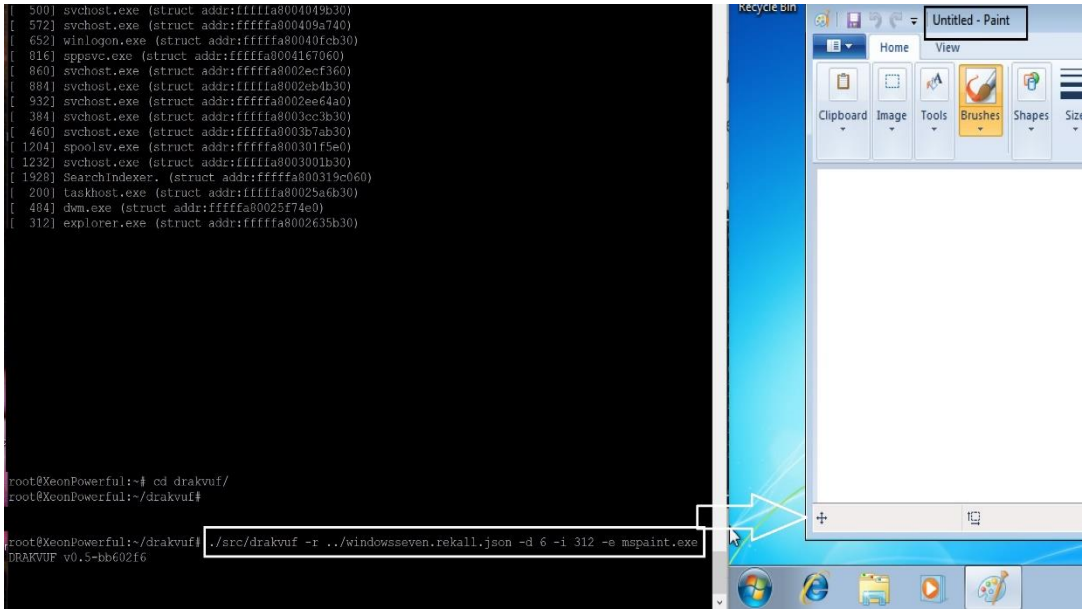
▲ VMI process injection into Windows 7 x64

List guest OS processes

```
root@XeonPowerful:~# vmi-process-list windowsseven
\\\\Process listing for VM windowsseven (id=6)
[    4] System (struct addr:fffffa8002366890)
[  228] smss.exe (struct addr:fffffa8003507040)
[  308] csrss.exe (struct addr:fffffa8003394820)
[  344] wininit.exe (struct addr:fffffa800333eb30)
[  352] csrss.exe (struct addr:fffffa8003341550)
[  392] services.exe (struct addr:fffffa8003fc5490)
[  404] lsass.exe (struct addr:fffffa8003fc9b30)
[  412] lsm.exe (struct addr:fffffa8003fd2b30)
[  500] svchost.exe (struct addr:fffffa8004049b30)
[  572] svchost.exe (struct addr:fffffa800409a740)
[  652] winlogon.exe (struct addr:fffffa80040fcb30)
[  816] sppsvc.exe (struct addr:fffffa8004167060)
[  860] svchost.exe (struct addr:fffffa8002ecf360)
[  884] svchost.exe (struct addr:fffffa8002eb4b30)
[  932] svchost.exe (struct addr:fffffa8002ee64a0)
[  384] svchost.exe (struct addr:fffffa8003cc3b30)
[  460] svchost.exe (struct addr:fffffa8003b7ab30)
[ 1204] spoolsv.exe (struct addr:fffffa800301f5e0)
[ 1232] svchost.exe (struct addr:fffffa8003001b30)
[ 1928] SearchIndexer. (struct addr:fffffa800319c060)
[  200] taskhost.exe (struct addr:fffffa80025a6b30)
[  484] dwm.exe (struct addr:fffffa80025f74e0)
[  312] explorer.exe (struct addr:fffffa8002635b30)
```

DRAKVUF Demo

▲ VMI process injection into Windows 7 x64



DRAKVUF Demo

▲ Configure socketmon plugin (monitors the usage of TCP and UPD sockets for Windows guests)

```
apt-get install python-construct python-pefile
```

```
git clone https://github.com/moyix/pdbparse
```

```
cd pdbparse/
```

```
python setup.py build
```

```
cd examples/
```

```
./symchk.py -e tcpip.sys
100%
()
Saved symbols to tcpip.pd_
Extracting cabinet: tcpip.pd_
  extracting tcpip.pdb

All done, no errors.
```

```
rekall parse_pdb tcpip.pdb > tcpip.json
```

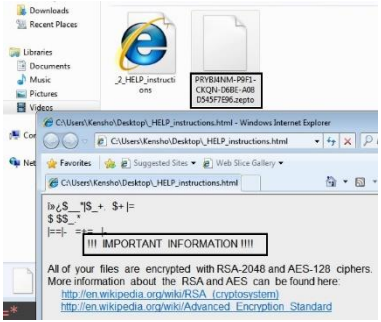
DRAKVUF Demo

▲ Start monitoring the sockets

```
root@XeonPowerful:~/drakvuf# xl list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0  1023     8     r-----    1039.9
windowsseven                                 4  3000     2     -b----     132.0
root@XeonPowerful:~/drakvuf# ./src/drakvuf  -d windowsseven -r /root/windowsseven.rekall.json -T /root/pdbparse/examples/tcpip.json
DRAKVUF v0.5-bb602f6
[POOLMON] vCPU:0 CR3:0x187000,(null) SessionID:-1 UHUB (type: NonPagedPool, size: 72): <unknown>,Universal Serial Bus Hub
[POOLMON] vCPU:0 CR3:0x187000,(null) SessionID:-1 Io   (type: NonPagedPool, size: 64): nt!io,general IO allocations
[SYSCALL] vCPU:1 CR3:0x5bc48000,svchost.exe SessionID:0 ntoskrnl.exe!NtReleaseWorkerFactoryWorker Arguments: 1
        IN HANDLE WorkerFactoryHandle: 0x90
[SYSCALL] vCPU:1 CR3:0x5bc48000,svchost.exe SessionID:0 ntoskrnl.exe!NtWaitForMultipleObjects Arguments: 5
        IN ULONG Count: 0x25
        IN HANDLE Handles[]: 0x159f60
        IN WAIT_TYPE WaitType: 0x1
        IN BOOLEAN Alertable: 0x1
        IN PLARGE_INTEGER Timeout: 0x0
```

DRAKVUF Demo

▲ Zepto ransomware




```
[FILETRACER] VCPU:1 CR3:0x314f1000,explorer.exe SessionID:1 \??\C:\Users\Kensho\Music\ZEPTO.exe
[SYSCALL] vCPU:1 CR3:0x314f1000,explorer.exe SessionID:1 ntoskrnl.exe!NtCreateFile Arguments: 11
IN HANDLE FileHandle: 0xb70 -> '\Users\Kensho\Music\ZEPTO.exe'
[FILETRACER] VCPU:1 CR3:0x314f1000,explorer.exe SessionID:1 \??\C:\Users\Kensho\Music\ZEPTO.exe
[SYSCALL] vCPU:1 CR3:0x314f1000,explorer.exe SessionID:1 ntoskrnl.exe!NtOpenFile Arguments: 6
[SYSCALL] vCPU:1 CR3:0x314f1000,explorer.exe SessionID:1 ntoskrnl.exe!NtQueryInformationFile Arguments: 5
      IN HANDLE FileHandle: 0xb70 -> '\Users\Kensho\Music\ZEPTO.exe'
[FILETRACER] VCPU:1 CR3:0x8107000,ZEPTO.exe SessionID:1 \??\C:\Windows\SysWOW64\CRYPTSP.dll
[SYSCALL] vCPU:1 CR3:0x8107000,ZEPTO.exe SessionID:1 ntoskrnl.exe!NtOpenFile Arguments: 6
```

⬇ Files are being encrypted

```
[FILETRACER] VCPU:0 CR3:0x9848000,SearchProtocol SessionID:0 \??\C:\Users\Public\Pictures\Sample Pictures\PRYBJ4NM-P9F1-CKQN-B6C5-38B2BE8C3D47.zepto
[SYSCALL] vCPU:0 CR3:0x9848000,SearchProtocol SessionID:0 ntoskrnl.exe!NtCreateFile Arguments: 11
```

References

- Github

https://github.com/tklengyel/drakvuf

- Drakvuf

https://drakvuf.com/

- Xen 4.9

https://wiki.xenproject.org/wiki/Category:Xen_4.9