



# DRAKVUF

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

**./ DRAKVUF™**

Black-box Binary Analysis System

# 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 bc bin86 gawk bridge-utils iproute libcurl3 libcurl4-openssl-dev bzip2 pciutils-dev build-essential make gcc clang 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 e2fslib-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-dev 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-github
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	VCPU	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 = "Windows7"
oslabel="drakvuf:vm_xdrakvuf_dom0_r"
maxmem = 3000
memory = 3000
vcpus = 2
maxvcpus = 2
builder = "hvm"
boot = "cd"
hap = 1
smp = 1
on_poweroff = "destroy"
on_reset = "destroy"
on_crash = "destroy"
vnc=1
vnclisten="0.0.0.0"
usb = 1
usbdevice = "tablet"
attache = 2
shadow_memory = 16
soundhw="hda"
vifs = [ 'type=ioemu,model=e1000,bridge=xenbr0,mac=10:60:4b:7e:ed:c7,ip=192.168.86.80' ]
disk = [ 'phy:/dev/xvdb1/LogicVolume,hda,v', 'file:/root/windows7.iso,hdr: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

```

root@XeonPowerLinux:~/drakvuf/1/bvml/examples# ./vml-win-guid name windowsx86vyn
Windows Kernel found @ 0xd217000
Version: 64-bit Windows 7
PE GUID: 1c9e251a5a5000
PDB GUID: 3844dbb9201495/be7aa4a2c201301a2
Kernel filename: ntkrnlmp.pdb
Multi-processor without PAE

MachineName:
* OF symbolName 24.
* of symbolName 0.
VirtualAddr: 100004000.
VirtualSize: 100004000.
Optional header size: 240.
Optional header: 0x0.
Section 1: 11040.
Section 2: 11040.
Section 3: 11040.
Section 4: 11040.
Section 5: 11040.
Section 6: 11040.
Section 7: 11040.
Section 8: 11040.
Section 9: 11040.
Section 10: 11040.
Section 11: 11040.
Section 12: 11040.
Section 13: 11040.
Section 14: 11040.
Section 15: 11040.
Section 16: 11040.
Section 17: 11040.
Section 18: 11040.
Section 19: 11040.
Section 20: 11040.
Section 21: 11040.
Section 22: 11040.
Section 23: 11040.
Section 24: 11040.
Section 25: 11040.
Section 26: 11040.
Section 27: 11040.
Section 28: 11040.
Section 29: 11040.
Section 30: 11040.
Section 31: 11040.
Section 32: 11040.
Section 33: 11040.
Section 34: 11040.
Section 35: 11040.
Section 36: 11040.
Section 37: 11040.
Section 38: 11040.
Section 39: 11040.
Section 40: 11040.
Section 41: 11040.
Section 42: 11040.
Section 43: 11040.
Section 44: 11040.
Section 45: 11040.
Section 46: 11040.
Section 47: 11040.
Section 48: 11040.
Section 49: 11040.
Section 50: 11040.
Section 51: 11040.
Section 52: 11040.
Section 53: 11040.
Section 54: 11040.
Section 55: 11040.
Section 56: 11040.
Section 57: 11040.
Section 58: 11040.
Section 59: 11040.
Section 60: 11040.
Section 61: 11040.
Section 62: 11040.
Section 63: 11040.
Section 64: 11040.
Section 65: 11040.
Section 66: 11040.
Section 67: 11040.
Section 68: 11040.
Section 69: 11040.
Section 70: 11040.
Section 71: 11040.
Section 72: 11040.
Section 73: 11040.
Section 74: 11040.
Section 75: 11040.
Section 76: 11040.
Section 77: 11040.
Section 78: 11040.
Section 79: 11040.
Section 80: 11040.
Section 81: 11040.
Section 82: 11040.
Section 83: 11040.
Section 84: 11040.
Section 85: 11040.
Section 86: 11040.
Section 87: 11040.
Section 88: 11040.
Section 89: 11040.
Section 90: 11040.
Section 91: 11040.
Section 92: 11040.
Section 93: 11040.
Section 94: 11040.
Section 95: 11040.
Section 96: 11040.
Section 97: 11040.
Section 98: 11040.
Section 99: 11040.
Section 100: 11040.
Section 101: 11040.
Section 102: 11040.
Section 103: 11040.
Section 104: 11040.
Section 105: 11040.
Section 106: 11040.
Section 107: 11040.
Section 108: 11040.
Section 109: 11040.
Section 110: 11040.
Section 111: 11040.
Section 112: 11040.
Section 113: 11040.
Section 114: 11040.
Section 115: 11040.
Section 116: 11040.
Section 117: 11040.
Section 118: 11040.
Section 119: 11040.
Section 120: 11040.
Section 121: 11040.
Section 122: 11040.
Section 123: 11040.
Section 124: 11040.
Section 125: 11040.
Section 126: 11040.
Section 127: 11040.
Section 128: 11040.
Section 129: 11040.
Section 130: 11040.
Section 131: 11040.
Section 132: 11040.
Section 133: 11040.
Section 134: 11040.
Section 135: 11040.
Section 136: 11040.
Section 137: 11040.
Section 138: 11040.
Section 139: 11040.
Section 140: 11040.
Section 141: 11040.
Section 142: 11040.
Section 143: 11040.
Section 144: 11040.
Section 145: 11040.
Section 146: 11040.
Section 147: 11040.
Section 148: 11040.
Section 149: 11040.
Section 150: 11040.
Section 151: 11040.
Section 152: 11040.
Section 153: 11040.
Section 154: 11040.
Section 155: 11040.
Section 156: 11040.
Section 157: 11040.
Section 158: 11040.
Section 159: 11040.
Section 160: 11040.
Section 161: 11040.
Section 162: 11040.
Section 163: 11040.
Section 164: 11040.
Section 165: 11040.
Section 166: 11040.
Section 167: 11040.
Section 168: 11040.
Section 169: 11040.
Section 170: 11040.
Section 171: 11040.
Section 172: 11040.
Section 173: 11040.
Section 174: 11040.
Section 175: 11040.
Section 176: 11040.
Section 177: 11040.
Section 178: 11040.
Section 179: 11040.
Section 180: 11040.
Section 181: 11040.
Section 182: 11040.
Section 183: 11040.
Section 184: 11040.
Section 185: 11040.
Section 186: 11040.
Section 187: 11040.
Section 188: 11040.
Section 189: 11040.
Section 190: 11040.
Section 191: 11040.
Section 192: 11040.
Section 193: 11040.
Section 194: 11040.
Section 195: 11040.
Section 196: 11040.
Section 197: 11040.
Section 198: 11040.
Section 199: 11040.
Section 200: 11040.
Section 201: 11040.
Section 202: 11040.
Section 203: 11040.
Section 204: 11040.
Section 205: 11040.
Section 206: 11040.
Section 207: 11040.
Section 208: 11040.
Section 209: 11040.
Section 210: 11040.
Section 211: 11040.
Section 212: 11040.
Section 213: 11040.
Section 214: 11040.
Section 215: 11040.
Section 216: 11040.
Section 217: 11040.
Section 218: 11040.
Section 219: 11040.
Section 220: 11040.
Section 221: 11040.
Section 222: 11040.
Section 223: 11040.
Section 224: 11040.
Section 225: 11040.
Section 226: 11040.
Section 227: 11040.
Section 228: 11040.
Section 229: 11040.
Section 230: 11040.
Section 231: 11040.
Section 232: 11040.
Section 233: 11040.
Section 234: 11040.
Section 235: 11040.
Section 236: 11040.
Section 237: 11040.
Section 238: 11040.
Section 239: 11040.
Section 240: 11040.
Section 241: 11040.
Section 242: 11040.
Section 243: 11040.
Section 244: 11040.
Section 245: 11040.
Section 246: 11040.
Section 247: 11040.
Section 248: 11040.
Section 249: 11040.
Section 250: 11040.
Section 251: 11040.
Section 252: 11040.
Section 253: 11040.
Section 254: 11040.
Section 255: 11040.
Section 256: 11040.
Section 257: 11040.
Section 258: 11040.
Section 259: 11040.
Section 260: 11040.
Section 261: 11040.
Section 262: 11040.
Section 263: 11040.
Section 264: 11040.
Section 265: 11040.
Section 266: 11040.
Section 267: 11040.
Section 268: 11040.
Section 269: 11040.
Section 270: 11040.
Section 271: 11040.
Section 272: 11040.
Section 273: 11040.
Section 274: 11040.
Section 275: 11040.
Section 276: 11040.
Section 277: 11040.
Section 278: 11040.
Section 279: 11040.
Section 280: 11040.
Section 281: 11040.
Section 282: 11040.
Section 283: 11040.
Section 284: 11040.
Section 285: 11040.
Section 286: 11040.
Section
```

# DRAKVUF Installation

## ▲ Create the LibVMl 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

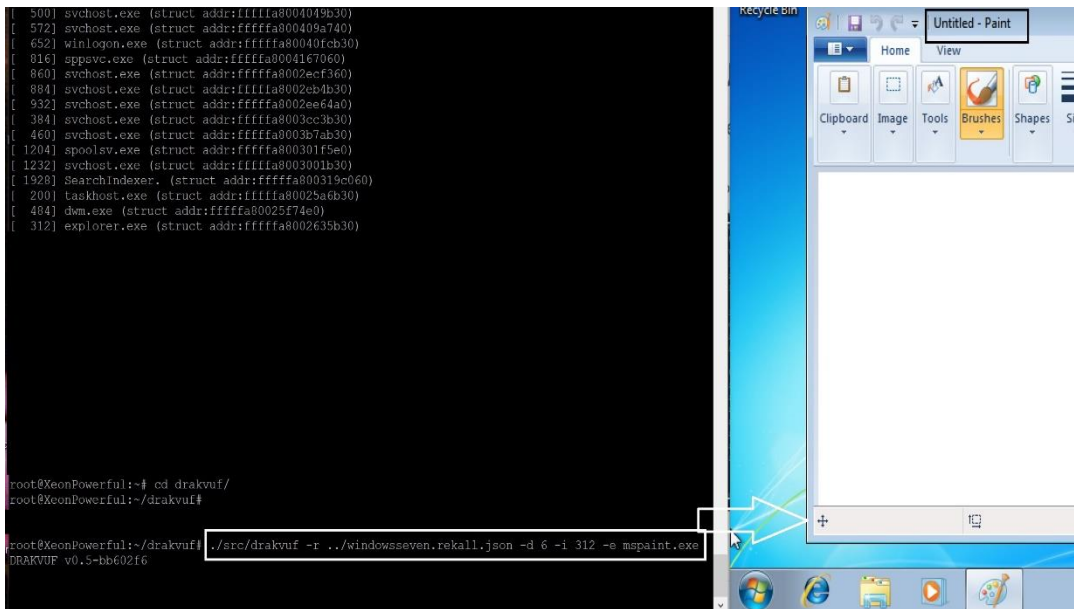
## ▲ VMI process injection into Windows 7 x64

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
```

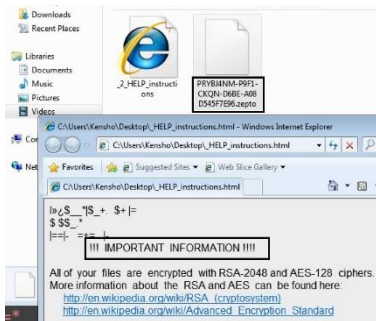
# DRAKVUF Demo

## ▲ 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](https://wiki.xenproject.org/wiki/Category:Xen_4.9)