LO-PHI: Low-Observable Physical Host Instrumentation for Malware Analysis

MAR 17TH, 2016

论文下载

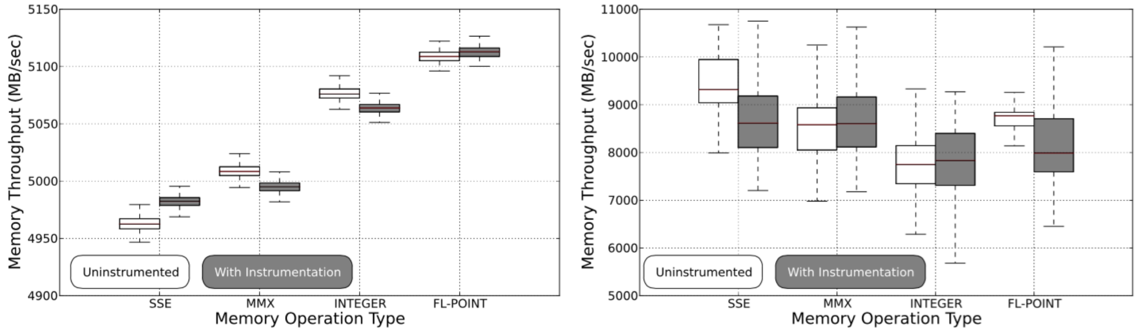
1. 目标: 利用物理方法实现对恶意软件行为分析的分析环境，解决恶意软件会对自己是否处于分析环境中进行检查并提前退出这一个问题。
2. 通过在`system under test`上插拔一些硬件的方式，在线监控物理主机的一些状态(内存，硬盘)，获得的原始数据会使用已有的开源软件`Volatility`和`Sleuthkit`填补信息
3. 已有一些获取裸机运行时物理内存的方法，本文主要开发了监控SATA硬盘的物理方法。并结合之前人的工作将各种技术整合成为一个可用的分析系统
4. 系统实现:
   - 物理实现
     - 内存在线读取: 一块Xilinx ML507插在的PCIe接口上的板子，因为PCIe设备是可以读取机子上

    的全部内存的，而且这个技术一直在之前的balabala文献中使用，而且PCIe的速度非常快

    - Disk: 一个有连个SATA接口的板子ML507，做主机和硬盘之间的中间人，并将每一个数据用千兆口网卡以UDP的形式发出出来
    - 做了个假键盘，和一个不太好的鼠标，然后就可以远程控制这个物理主机，并提到有一个现成的设备可以帮助完成类似的工作
    - 最后，因为不像虚拟机可以随时恢复镜像，所以系统是使用PXE，然后网络启动的，这样重置设备只需要重启电脑就可以了
- 对应的作者还部署了基于虚拟化技术实现的版本:
    - 内存读取，硬盘数据，交互，状态恢复

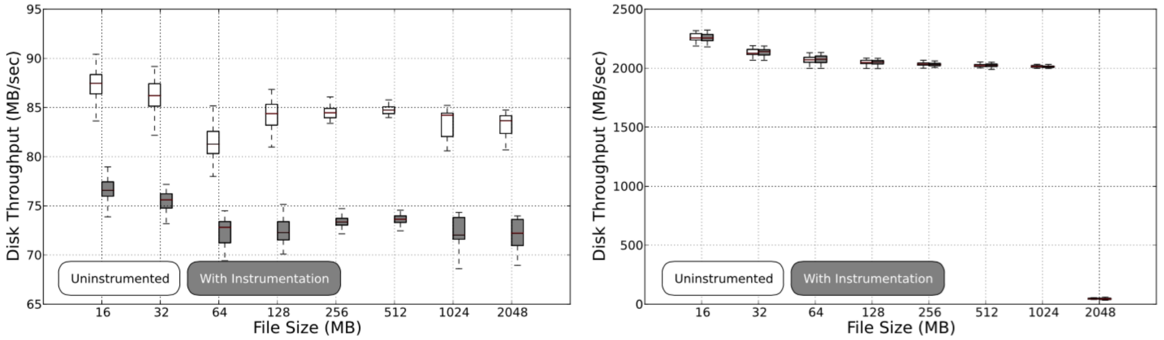
5 性能测试:

- 内存: RAMSpeed做测试
- 硬盘： IOZone



(a) Physical machine (Polling at 14MB/sec)

(b) Virtual machine (Polling at 160MB/sec)



(a) File writes

(b) File reads

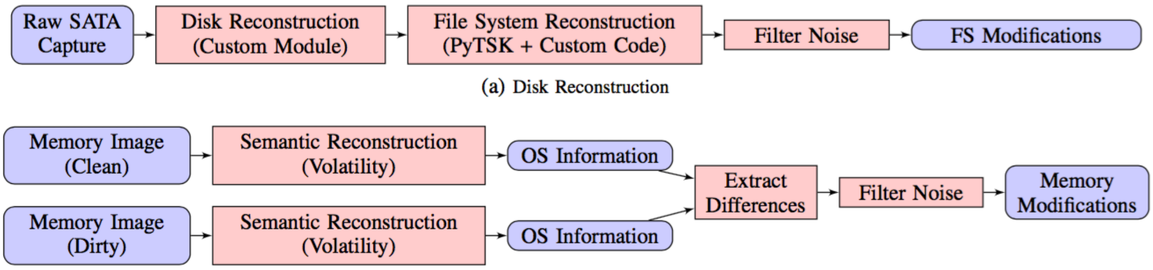
PSI可对程序及其所有用到的共享库插桩。输入binary，输出插桩后的Bianry。插桩可以在执行前，也可以在程序执行过程中。

Limitations

- DMA是可以禁用的，在有IOMMU的机子上。不懂。。。
- 内存是一点一点读的，系统不能获得一个时刻的内存完整dump，而是分段读出来的，因此dump过程内存可能已经被被修改。不过实践中很少出这种问题并不影响分析的结果。
- 文件系统缓存导致对硬盘的记录不准确，这个问题很少出现。
- “the network policies within our organization currently forbid us from running these malware samples on the live Internet. ”所以所有的分析都是没有网的。。。 **实验:**
- 分析流程如下图

```
# Reset our disk using PXE
machine.machine_reset()
machine.power_on()
# Wait for OS to appear on network
while not machine.network_get_status():
    time.sleep(1)
# Allow time for OS to continue loading
time.sleep(OS_BOOT_WAIT)
# Start disk capture
disk_tap.start()
# Send key presses to download binary
machine.keypress_send(ftp_script)
# Dump memory (clean)
machine.memory_dump(memory_file_clean)
# Start collection network traffic
network_tap.start()
# Get a list of current visible buttons
button_clicker.update_buttons()
# Start our binary and click any buttons
machine.keypress_send('SPECIAL:RETURN')
# Move our mouse to imitate a human
machine.mouse_wiggle(True)
# Allow binary to execute (Initial)
time.sleep(MALWARE_START_TIME)
# Dump memory (interim)
machine.memory_dump(memory_file_interim)
# Take a screenshot (Before clicking buttons)
machine.screenshot(screenshot_one)
# Click any new buttons that appeared
button_clicker.click_buttons(new_only=True)
# Allow binary to execute (3 min total)
time.sleep(MALWARE_EXECUTION_TIME-elapsed_time)
# Take a final screenshot
machine.screenshot(screenshot_two)
# Dump memory (Dirty)
machine.memory_dump(memory_file_dirty)
# Shutdown Machine
machine.power_shutdown()
```

- 在恶意软件运行运行前后，分别获取全部物理内存，然后恢复语义信息之后进行diff，硬盘也类似。

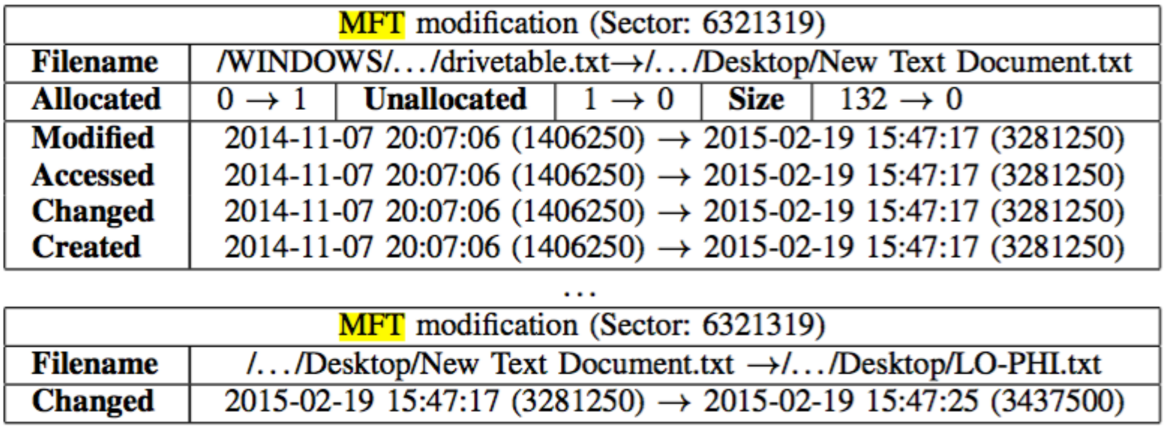


(a) Disk Reconstruction

(b) Memory Reconstruction

Fig. 5: Binary analysis workflow. (Rounded nodes represent data and rectangles represent data manipulation.)

- 语义恢复
  - 用尽了volatility里的各种功能，列进程，端口,ssdt,service
  - 重放硬盘操作，恢复NTFS的Master File Table, 然后就有了如下的结果



MFT modification (Sector: 6321319)					
Filename	/WINDOWS/.../drivetable.txt→/.../Desktop/New Text Document.txt				
Allocated	0 → 1	**Unallocated**	1 → 0	**Size**	132 → 0
Modified	2014-11-07 20:07:06 (1406250) → 2015-02-19 15:47:17 (3281250)				
Accessed	2014-11-07 20:07:06 (1406250) → 2015-02-19 15:47:17 (3281250)				
Changed	2014-11-07 20:07:06 (1406250) → 2015-02-19 15:47:17 (3281250)				
Created	2014-11-07 20:07:06 (1406250) → 2015-02-19 15:47:17 (3281250)				

...

MFT modification (Sector: 6321319)	
Filename	/.../Desktop/New Text Document.txt →/.../Desktop/LO-PHI.txt
Changed	2015-02-19 15:47:17 (3281250) → 2015-02-19 15:47:25 (3437500)

- 与一些与现有基于虚拟化技术实现的恶意软件行为分析引擎进行了比较，证明有很多样本会检测是否在虚拟环境中而不展现出恶意行为，但无法检测自身处于作者构建的物理分析环境中。

总结：

本文综合了之前的一些对PC进行物理分析内存取证和硬盘取证的手段，并形成了一整套的分析手段。