SIFT Workstation 2.0: SANS Investigative Forensic Toolkit

By Russ McRee – ISSA member, Puget Sound (Seattle), USA Chapter

Prerequisites

VMWare for the SIFT 2.0 VMWare Appliance

Similar Projects Helix 3

Forensicators, if you haven't yet made use of SIFT Workstation, now is the time!

The SANS Investigative Forensic Toolkit (SIFT) Workstation 2.0 is a Linux distribution that is preconfigured for forensic investigations. Created as part of Rob Lee's SANS 508 track, "Computer Forensic Investigations and Incident Response," version 2.0 is a recent release and includes all the tools a forensic analyst would require to conduct a thorough system investigation. I particularly favor it for memory analysis – grab a memory image from your victim system; pull it back to your SIFT VM; and get down to business in no time flat.

For top notch insight on Windows incident response, including memory analysis, be sure to read Harlan Carvey's blog of the same name[1]

Installing and configuring SIFT

SIFT 2.0 can be utilized via your preferred version of VMWare (I run it regularly via VMWare Server 2.0.2). You can also download it as a DVD.iso should you choose to install it as a standalone workstation. Using the VMWare appliance is recommended as you can run multiple instances specific to each investigation you're conducting, enhanced by the ability to take snapshots to avoid evidence or findings corruption of any kind (roll back to last known good state).

Using SIFT 2.0

To best exemplify the raw forensicating power that is SIFT 2.0, we'll run through as current and relevant a scenario as can be imagined (only names and places have been changed to protect the innocent).

You are an information security analyst with a large corporate network and many users of varying security awareness and savvy. Your duties include incident response and forensic analysis, particularly where malware infections are in scope.

A user emails you indicating that he received a shortened URL over instant messaging from a trusted contact; specifically, tinyurl.com/y6v689z. But when the user clicked the URL, uh-oh: "it gave me a file and I clicked it. It didn't seem right so I emailed you." The user indicates that "nothing else weird seems to be going on" with his system, but you know that immediate investigation is required.

Sound like a familiar conversation? I imagine we've all had a similar chat at one time or another. ;-)

You'll take a slightly different approach from here. Where you might normally conduct some sort of live incident response on the actual victim system, instead you intend to go straight for the memory imaging option and analysis in order to facilitate a rapid determination using SIFT 2.0.

You take a victim memory image with the MoonSols Windows Memory Toolkit[2] (community version), that includes Matthew Suiche's `win32dd.exe`. The MoonSols kit is worth a *toolsmith* write-up all by itself, but that's for the future. That being said, the MoonSols Memory Toolkit is one of those indispensable toolsets I simply can't live without; be sure to add it to your kit too.

PSEXEC a shell to the victim system and run

`win32dd /f \\SMBCaseFolder\VictimMem.img`, or run it against a remote machine from your Windows workstation with the `/t` switch. MoonSols includes `win64dd.exe`, which is increasingly more important as the predominance of 64bit hardware continues to grow in desktop PCs (it goes without saying in datacenters).

Once you've captured the memory image from the PC of your our wayward clicking victim, you can make use of SIFT 2.0 tooling to conduct analysis.

Note: I'll share the memory image I created for this exercise upon request; just email me if you'd like a copy.

For this scenario you're copying all relevant case files to `/home/sansforensics/Desktop/cases/toolsmith-Case`.

Next, you make use of Volatility, another framework deeply worthy of its own *toolsmith* coverage, and inherent to the SIFT 2.0 Workstation.

Volatility uses include:

- connscan: scan for connection objects
- files: print list of open files for each process
- hibinfo: convert hibernation file to linear raw image
- procdump: dump a process to an executable sample

1 http://windowsir.blogspot.com.

2 http://moonsols.com/blog/9-moonsols-windows-memory-toolkit.

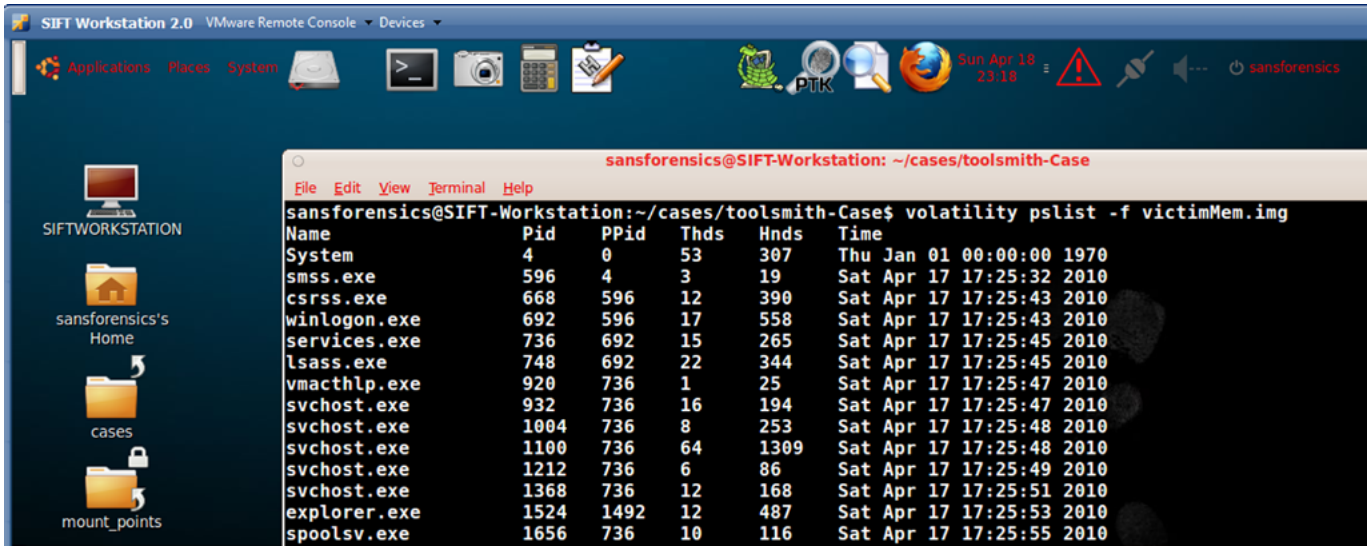


Figure 1 – Volatility spots suspicious processes

- pslist: print list of running processes
- regobjkeys: print list of open regkeys for each process
- sockets: print list of open sockets
- sockscan: scan for socket objects

There are numerous Volatility extended plugin options as well, but I promise you, for a down and dirty incident response, running any of the list above against a win32dd-captured memory image will likely allow you to make a prompt root cause determination.

As an example, running `volatility pslist -f victimMem.img` will print list of running processes (See Figure 1).

Now, I'm not a betting man, but I'll wager you lunch that `ftpqa.tif`, `vjkft.tif`, and `sqnfh.tif` are not legitimate Windows processes.

Run `volatility procdump -p 176 -f victimMem.img`. The –p switch passes the PID for `vjkft.tif` and dumps the process to an executable sample you can scan with your local antivirus client or feed to VirusTotal. Doing so with the resulting `executable.176.exe` revealed that `vjkft.tif` is really `Win32:Rootkit.gen`,[3] which also known as (a bit of a clue) `PWS-Banker.gen.b`. This reference indicates the prospect of a malware genus match to be confirmed later.

Using PID 176 as a marker we can also opt for `volatility regobjkeys -p 176 -f victimMem.img`, the results of which clearly indicate that this malicious process is manipulating Internet Explorer given \REGISTRY\MACHINE\SOFTWARE\MICROSOFT\INTERNET EXPLORER\MAIN\FEATURECONTROL\FEATURE_PROTOCOL_LOCKDOWN.

This is further confirmed via `volatility files -p 176 -f victimMem.img` which results in

```
File    \WINDOWS\system32\ieframe.dll
```

Have I mentioned how much I love Volatility?

`Ieframe.dll` is the Internet Explorer Browser UI Library (common malware fodder); the above mentioned registry key indicates the possibility that this malware is taking advantage of the fact that the victim PC has Internet Explorer Protected Mode and might be falling victim to a vulnerability in IE that allows information disclosure.[4]

Repeat `volatility procdump` and `volatility file` for the other suspicious PIDs represented by .tif-denoted processes as seen in Figure 1. When you submit the binary created from procdump for PID 3296 you note that Virustotal flagged[5] it as Delf amongst other things. Put Delf and PWS-Banker together and you've found yourself enough detail to reasonably conclude a Banload/Bancos infection. Yet, you should explore further…

PTK

SIFT 2.0 includes PTK,[6] "the alternative computer forensics framework" for The Sleuth Kit. You've likely heard of Autopsy, or used it; it's also installed should you prefer to make use of it.

PTK, in addition to the expected case and image management, also includes features similar to Volatility.

Assign a case name, and then add the memory image you've acquired, selecting DD for acquisition type, then RAM dump and the appropriate time zone. You can provide additional details as you see fit, including calculated hashes should you need to validate process and image integrity.

Click the added image name and select Analyze image (See Figure 2).

You ran the Volatility procdump command for PID 3296 (`sqnfh.tif`) earlier; investigate it further with PTK.

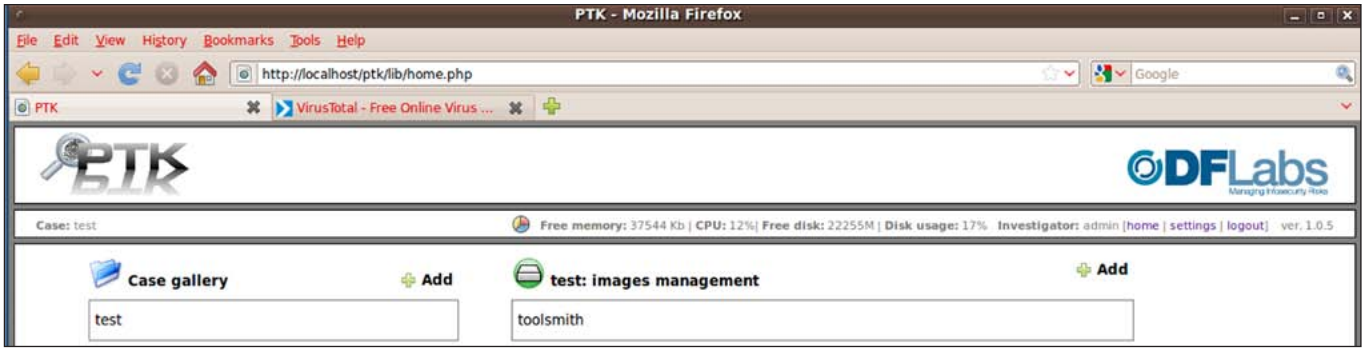
3 http://www.virustotal.com/analisis/fcd7bd4bc89a57f9bfec5bd05122b6b5b01a9e75684d2ec2aed226d54c87d01a-1271714424.

4 http://www.microsoft.com/technet/security/advisory/980088.mspx.

5 http://www.virustotal.com/analisis/feeca5c9da60bedd33ad915375269ce4740e6478bd8d20f52d2d2a858b73c8cf-1271913792.

6 http://ptk.dflabs.com.

Figure 2 – Add case and image to PTK



Select psscan from the analysis type dropdown menu and select Start.

Copy the offset `0x018f3d78` from PID 3296 findings to your clipboard.

Select vadinfo (Virtual Address Descriptors) from the analysis type dropdown menu and paste the offset to the form; click Start. Figure 3 clarifies the results.

Note that the FileObject path is `\WINDOWS\Fonts\sqnfh.tif`.

Now you have a physical file reference on the victim host to confirm against the Volatility procdump binary extracted from memory.

Keyword searches, bookmarks, and reporting all await you via PTK. Bookmark the vadinfo results for easy retrieval later and inclusion in your incident report.

You certainly have enough information to have the desktop technician wipe and reload your victim's PC, unless you need a full system image should a crime have occurred (financial accounts compromised).

Via only memory analysis you discovered TrojanDownloader:Win32/Banload.MC was indeed the culprit. You advise your victim to reset all passwords, and establish credit line alerts with the appropriate providers as Banload is a credentials stealer.

In conclusion

Keep in mind, this entire investigation was conducted without ever leaving your desk.

Now you know it's time for increased user awareness regarding shortened URLs and the false presumption of trust that all IM content is safe. Mash up shortened URLs, blind trust for IM contacts, and a "I click everything people send me" user, and you have yourself golden opportunities to expand your memory analysis and forensics skills with the help of SIFT 2.0. Get to it! ;-)

Cheers…until next month.

About the Author

Russ McRee, GCIH, GCFA, GPEN, CISSP, is team leader and senior security analyst for Microsoft's Online Services Security Incident Management team. As an advocate of a holistic approach to information security, Russ' website is holisticinfosec.org. Contact him at russ@holisticinfosec.org.

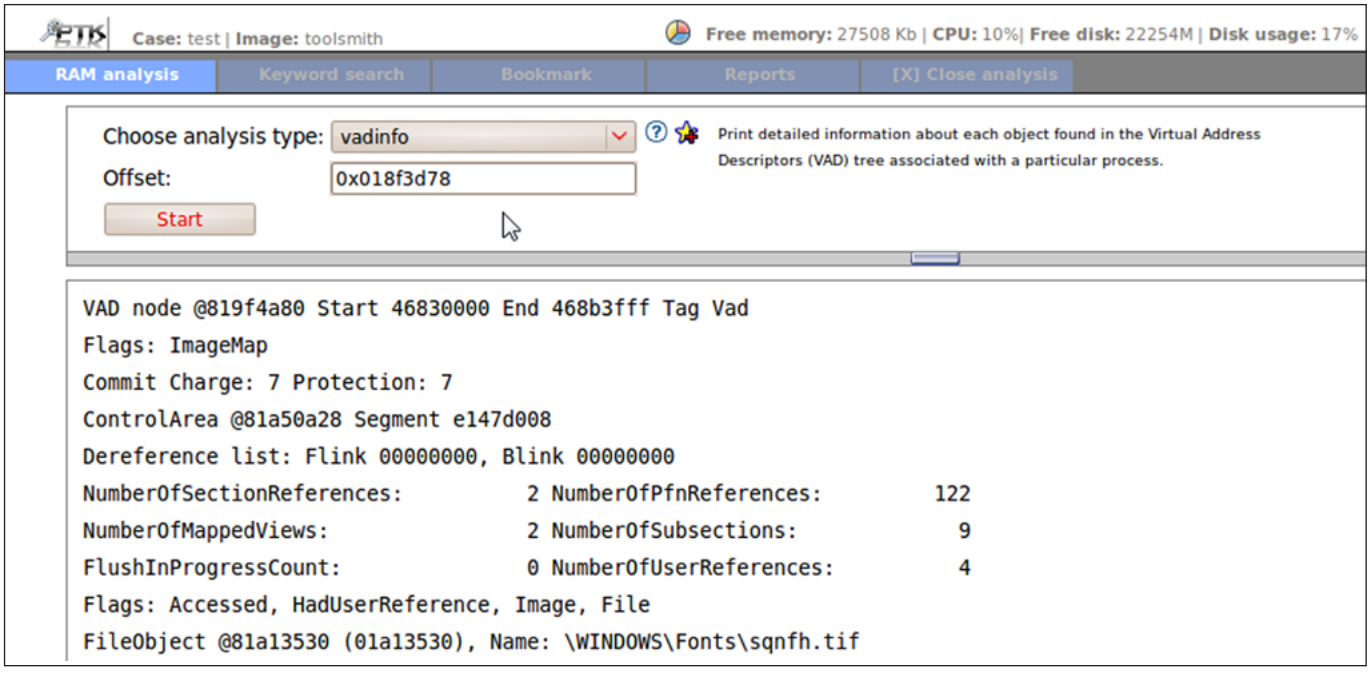


Figure 3 – Find the file path with VAD info