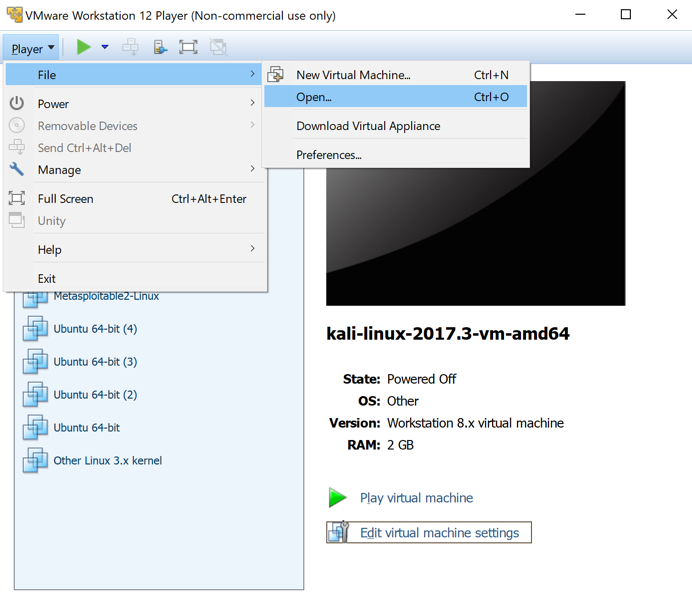
**SCY486 Lab exercise Five: RAM Analysis –Windows**

**Assigned: February 13, 201**

**Due Date: February 20,2019**

1. **Installing the Kali VM**
2. **Create a VM folder**
   1. Go to “This PC”. Find Documents folder, and look for Virtual Machines folder. In a there create a new folder called “YOUR Name VMs”.
   2. Download WinRAR from the Share. WinRAR is an unzipping tool.
   3. Download the Kali VM zipped file from the Lab1 Images folder from the Share (NOT THE iso) and unzip it into the directory you just created on the VMs folder.
3. **Starting your Kali Virtual Machine**

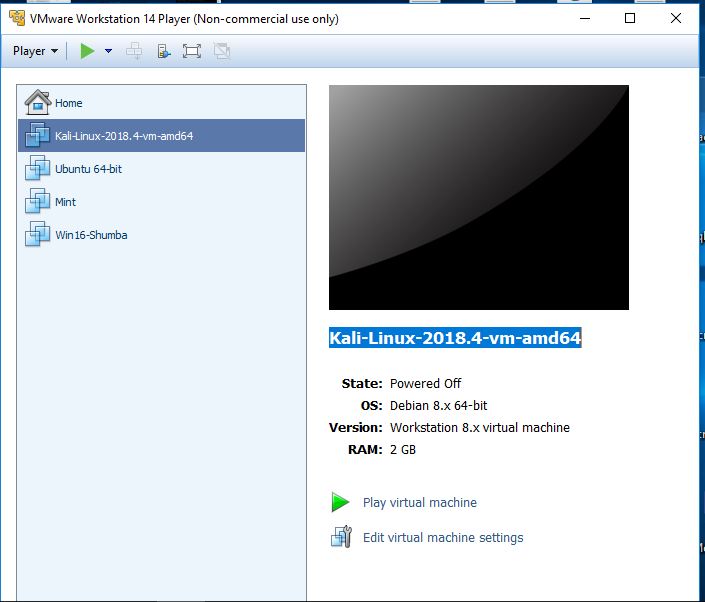
In VMware Player, at the top left, click Player, File, Open..., as shown below. Navigate to the Kali-Linux-2018-vm-amd64 folder and click on the file that displayed.



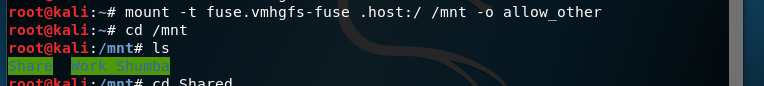
Log in to Kali with the username **root** and a password of **toor**.

**Now shutdown the Kali VM**

1. **Sharing folders between Windows and Kali**
2. Create a folder called “Images” on your host desktop to share with Kali.
3. Download a zipped file memdump.7z and cridex from the Memory\_Analysis folder, Shared drive into the created Images file.
4. With the Kali VM powered off, in VMWare Workstation, select the Kali VM and click on “Edit virtual machine settings”, as shown below:



1. Click options
2. Scroll down and highlight “Shared Folders”
3. Select “Always Enabled” on the right, if it is not selected.
4. Click “Add” and go use the Wizard until you select your “Images” folder.
5. Click “OK “ to finish the process.
6. Now start your Kali VM and open a terminal and type in the command given in the screenshot below:



1. Now, still within your VM, click “Places” at the top of the Kali screen, then select Computer, then mnt and you should be able to see the mounted folder there, from your Windows machine.
2. Now copy the zipped image from “Images” to your Desktop and extract it there.
3. **Analyzing your memory dump using volatility**

Assuming the path to your memory dump is **/root/Desktop/memdump.mem, l**et’s begin analyzing this RAM dump in volatility.

The first step in using volatility is to get the image information using the imageinfo and kdbgscan plugins. Please note the syntax for running these two commands.

volatility imageinfo -f /root/Desktop/memdump.mem

Sample command line:

#volatility -f <dumppath> --profile=Win10x64\_15063 plugin

Example:

#volatility -f /root/Desktop/memdump.mem --profile =<Win10Profile> plist

#volatility -f <image> --profile=Win7SP0x64 memdump -p 4 -D dump/

Below is the list of some plugins

1. **Imageinfo** shows basic information about the image, the operating system of the machine that was imaged, and when the image was made.
2. **pslist** prints all running processes. This command will display every running process on a system and can be used to prove that a specific process was open, or to look for a suspicious process in an investigation.
3. **kdbgscan** is designed to positively identify the correct profile and the correct KDBG address (if there happen to be multiple).
4. **consoles** shows the console commands that were recently executed on the Windows machine, beginning with the "csrss" process
5. **netscan** finds TCP/UDP endpoints and listeners, it display a list of active network connections. This would be useful in investigations to determine where traffic was coming from or going to, over which protocol, and which application was generating it.
6. **hivelist** find the physical addresses of registry hives and print the list of them.
7. **hashdump** dumps passwords hashes (LM/NTLM) from memory. This command can be used to display the hashed credentials for user accounts, and these hashes can then be used in other tools to determine their account passwords.
8. **pstree** prints the process list as a tree. This command displays the same information as pslist, only in tree form. This allows you to see which parent process everything belongs to.
9. psscan finds processes that were previously terminated or unlinked by a rootkit –can be used in an investigation to discover hidden malicious software such as key loggers or rootkits.
10. connections displays TCP connections that were active at the time of the memory acquisition, use the connections command.
11. memdump extracts all memory resident pages in a process into an individual file, use the memdump command.
12. procdump dump a process's executable, use the procdump command.

**memdump.mem image analysis**

**Use the plugins provided above to answer the below questions**

1. What was the operating system that the above dump came from?
2. Provide two suggested profiles for the image.
3. When was the dump created?
4. Now provide the more correct profile to use as given by the kdbgscan plugin
5. List at least 3 processes that were running when the dump was created, please pick the familiar ones.
6. Now, let us search if a specific process, “notepad” was running when the image was taken:

#volatility –f <image path> --profile=<Profile> pslist|grep notepad

Record the process id for notepad here:

1. Now we can dump the process, using the below command:

#volatility -f <image path> --profile= <Profile> memdump --dump-dir=./ -p <PID>

Provide the name of the process dump here:

1. Now we want to find the strings that were typed in notepad. If we know a single word that was typed, we can search for it, else we have to dump all the strings in the dump and read through and find any typed in strings

# strings -e l ./<PID for dumped process>.dmp | grep "Some"

Any interesting strings found?

1. Now let’s see if there were passwords in the memory when the image was taken.
2. Find the physical addresses of registry hives and print the list of them.

*#v*olatility hivelist -f <image path> --profile=<Profile>

We now have a list of where several key items are located in the memory dump. Next we will extract the password hashes from the memory dump. To do this it is necessary to find the starting memory locations for the SYSTEM and SAM keys.

1. Look in the output and copy the numbers in the first column that correspond to the SAM and SYSTEM locations. Then output the password hashes into a text file called hash.txt.

#volatility hashdump -f <image path> -–profile=<Profile> -y <SYSTEM START> -s <SAM START> > hashes.txt

The hash values in hash.txt file can now be cracked with any password cracking tool. You can go to https:// crackstation.net

Cridex memory dump:

1. What was the operating system that the above dump came from?
2. When was the dump created?
3. Now provide the more correct profile to use as given by the kdbgscan plugin
4. Look closely at the processes that were running when the dump was taken. Are there any suspicious processes?
5. Use the connections plugin to check the connections that were active at the time of the memory acquisition. Record your observation.
6. Use the “whois” to check any suspicious IP address connections.
7. Using the procdump plugin, dump any suspicious processes into an executable.

#volatility -f <image path> --profile= <Profile> procdump --dump-dir=./ -p <PID>

1. Copy the executable of the suspicious process and run it in Virus Total. Is it a malware?