**Progress test 3-2**

**Volatility**

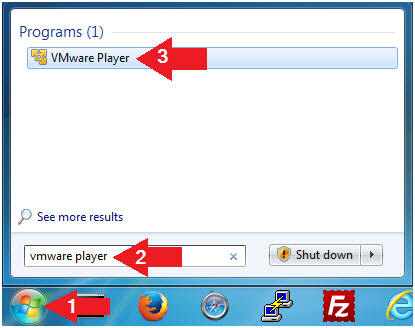
**Analyzing Memory Capture for Windows XP SP2**

**Section 0. Background Information**

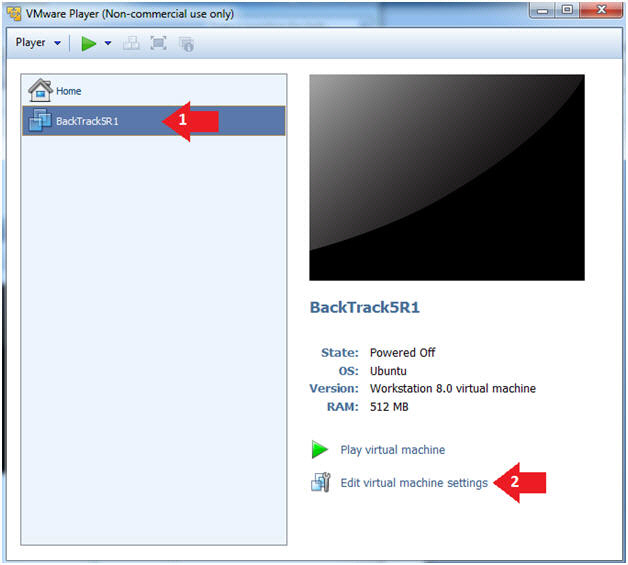
* **Volatility Overview**
  + <https://www.volatilesystems.com/>
  + The Volatility Framework is a completely open collection of tools, implemented in Python under the GNU General Public License, for the extraction of digital artifacts from volatile memory (RAM) samples. The extraction techniques are performed completely independent of the system being investigated but offer unprecedented visibility into the runtime state of the system. The framework is intended to introduce people to the techniques and complexities associated with extracting digital artifacts from volatile memory samples and provide a platform for further work into this exciting area of research.
* **Pre-requisite Labs**
  + Install Volatility Framework 2.0 on BackTrack5 Lab
  + Dump Window's Physical Memory to BackTrack5 Using Helix and NetCat

**Section 1: Login to BackTrack**

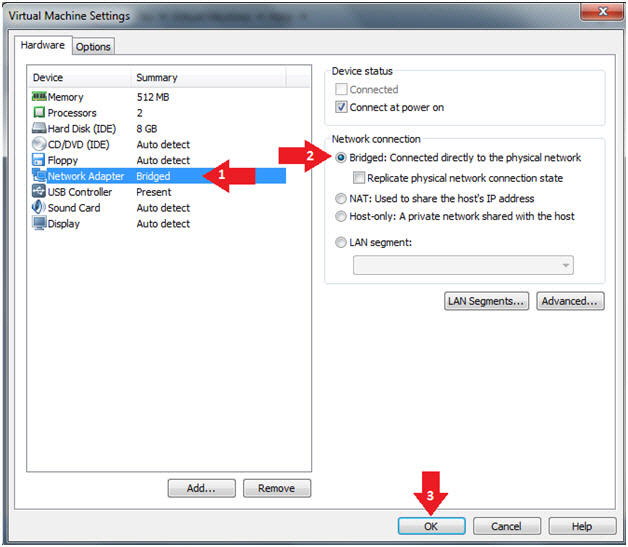
1. Open VMware Player on your (**Host Windows Machine**).
   * **Instructions:**
     1. Click the Start Button
     2. Type "vmware player" in the search box
     3. Click on VMware Player



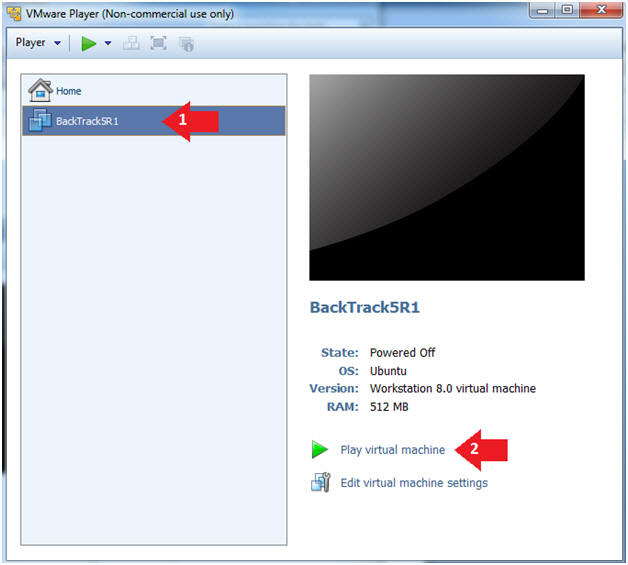
1. Edit the BackTrack5R1 VM
   * **Instructions:**
     1. Select BackTrack5R1 VM
     2. Click Edit virtual machine settings



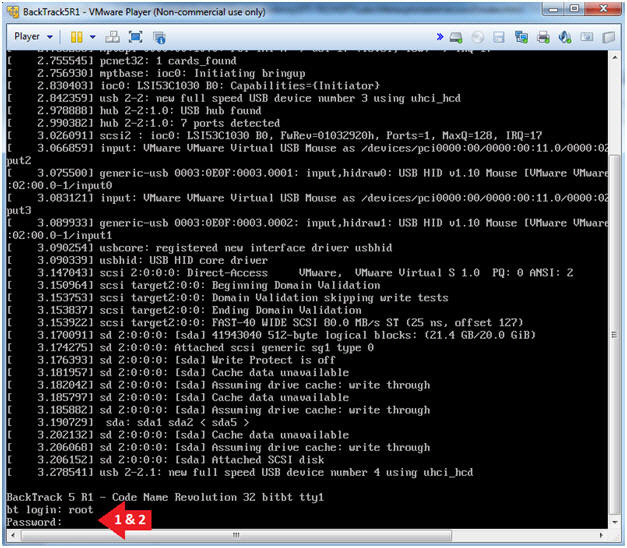
1. Edit Virtual Machine Settings
   * **Instructions:**
     1. Click on Network Adapter
     2. Click on the Bridged Radio button
     3. Click on the OK Button



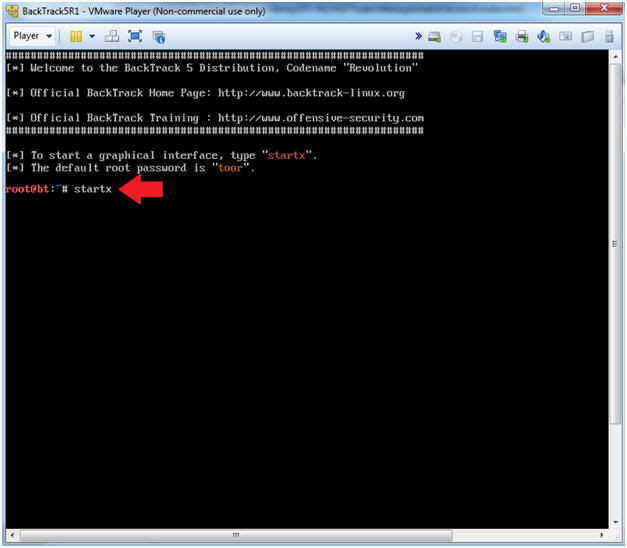
1. Play the BackTrack5R1 VM
   * **Instructions:**
     1. Click on the BackTrack5R1 VM
     2. Click on Play virtual machine



1. Login to BackTrack
   * **Instructions:**
     1. Login: root
     2. Password: toor or <whatever you changed it to>.

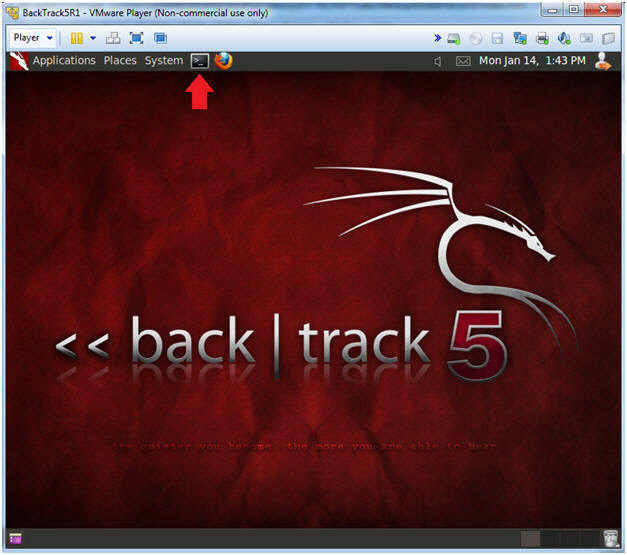


1. Bring up the GNOME
   * **Instructions**:
     1. Type startx



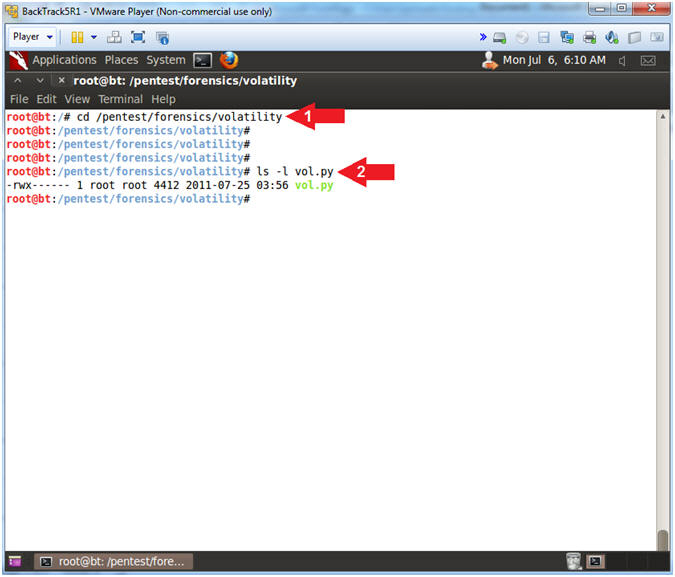
**Section 2: Bring up a console terminal**

1. Start up a terminal window
   * **Instructions**:
     1. Click on the Terminal Window

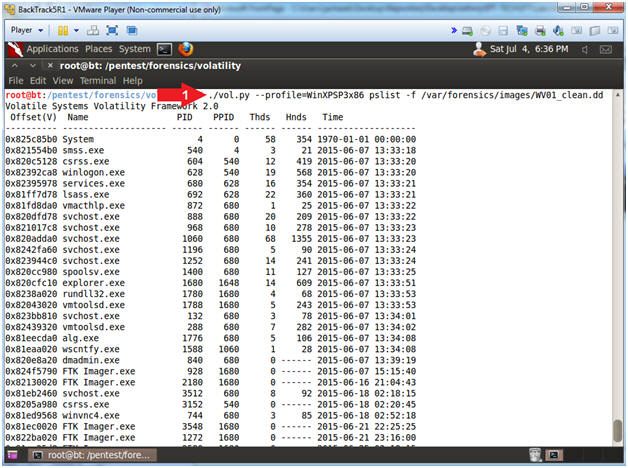


**Section 3: Using Volatility**

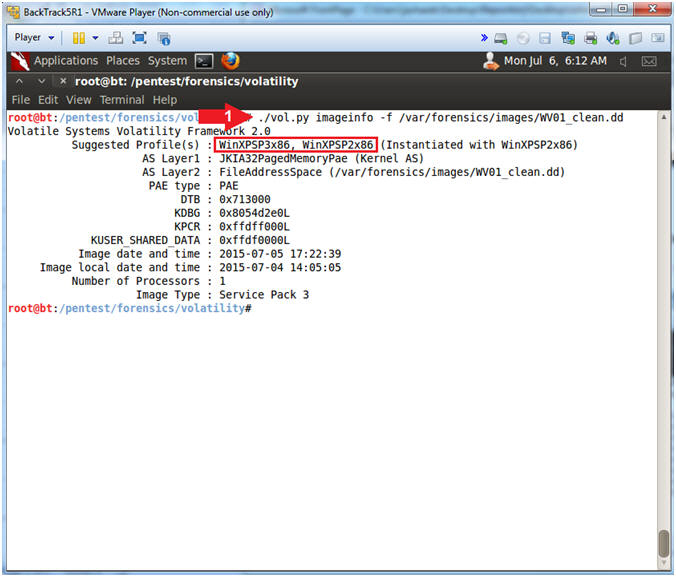
1. Navigate to Volatility
   * **Instructions**
     1. cd /pentest/forensics/volatility
     2. ls -l vol.py



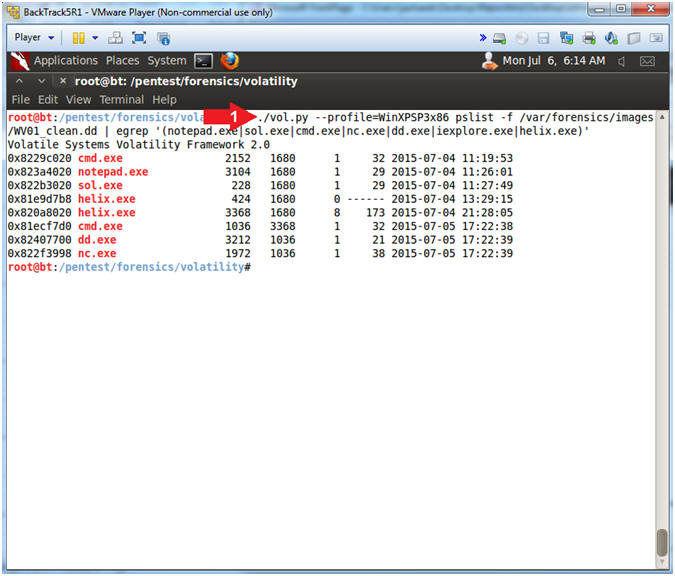
1. Obtain the image profile
   * **Instructions:**
     1. **./vol.py imageinfo -f /var/forensics/images/WV01\_clean.dd**
   * **Notes(FYI):**
     1. The Volatility Framework tries to guess and tell you what image profile to use.
     2. We know that our Server is Windows XP running SP2.
     3. Volatility suggest that we either use the profile WinXPSP3x86 or WinXPSP2x86.
     4. Also, we are running SP2 **we will use the WinXPSP3x86** which seems to have more complete profile than WinXPSP2x86.



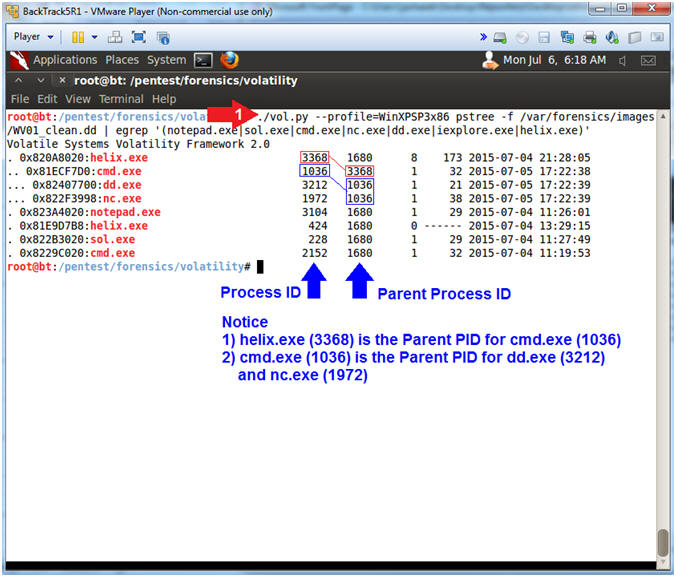
1. View Running Processes
   * **Instructions**
     1. **./vol.py --profile=WinXPSP3x86 pslist -f /var/forensics/images/WV01\_clean.dd**
   * **Notes(FYI):**
     1. This displays all the running process during the time we captured the image in [Helix Lesson 4](http://www.computersecuritystudent.com/FORENSICS/HELIX/lesson4/index.html).



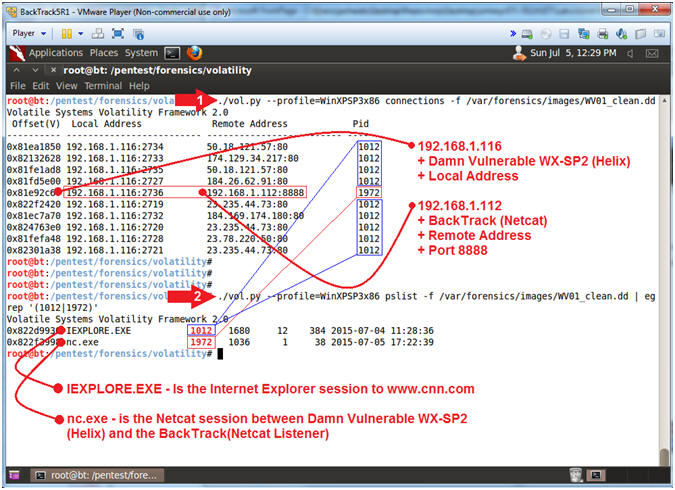
1. Searching for Specific Processes
   * **Instructions:**
     1. **./vol.py --profile=WinXPSP3x86 pslist -f /var/forensics/images/WV01\_clean.dd | egrep '(notepad.exe|sol.exe|cmd.exe|nc.exe|dd.exe|iexplore.exe|helix.exe)'**
   * **Notes(FYI):**
     1. egrep - Is like grep, but it lets you search for multiple strings.
     2. notepad.exe - Is the Notepad application.
     3. sol.exe - Is Solataire.
     4. cmd.exe - Is the Command Prompt that we started.
     5. nc.exe - Is NetCat which was started by Helix.
     6. dd.exe - Was started by Helix.  It made the memory image.
     7. iexplore.exe - Internet Explorer.
     8. helix.exe - Helix



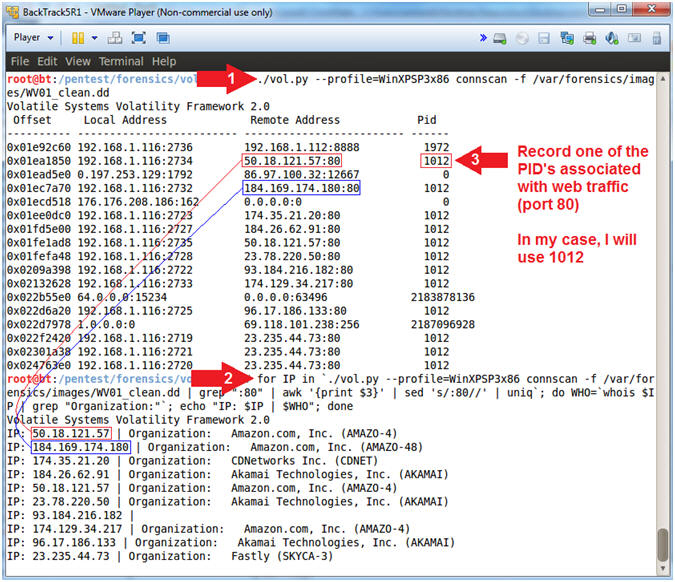
1. Associating Parent and Child Processes
   * **Instructions:**
     1. **./vol.py --profile=WinXPSP3x86 pstree -f /var/forensics/images/WV01\_clean.dd | egrep '(notepad.exe|sol.exe|cmd.exe|nc.exe|dd.exe|iexplore.exe|helix.exe)'**
   * **Notes(FYI):**
     1. pstree - Print process list as a tree
     2. Notice the two columns of numbers after executable.  The far left column is the Process ID (PID) for the executable.  The far right column is the Parent Process ID (PPID) for the executable.
     3. Notice that Helix's PID is 3368 and it is the PPID for the cmd.exe PID(1036).  Also, cmd.exe's PID(1036) is PPID for dd.exe(3212) and nc.exe(1972).  Thus, the Helix is the PPID for not only cmd.exe, but also, dd.exe and nc.exe.
     4. These PIDs will be different in your case!!!



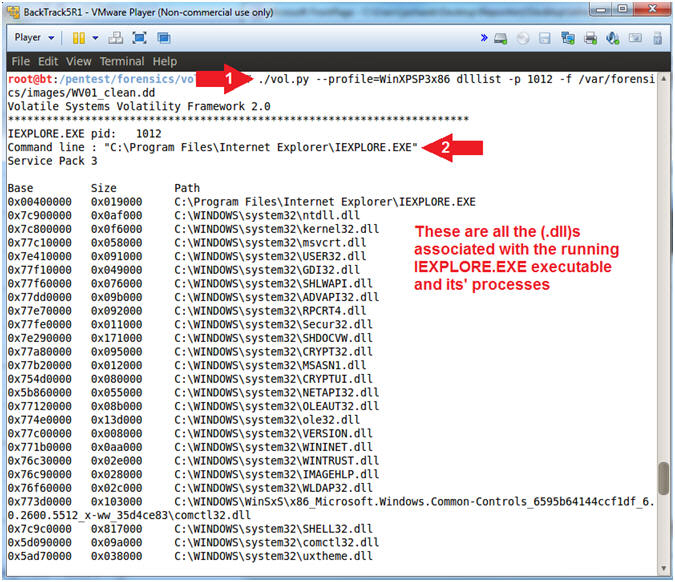
1. View Network Connections and Tie to Running Processes
   * **Instructions:**
     1. **./vol.py --profile=WinXPSP3x86 connections -f /var/forensics/images/WV01\_clean.dd**
        + This command lets you view **all open** Network Connections.
     2. **./vol.py --profile=WinXPSP3x86 pslist -f /var/forensics/images/WV01\_clean.dd | egrep '(1012|1972)'**
        + This command lets you search the process list for all the network connections.
   * **Note(FYI):**
     1. These numbers will be different in your case!!!
     2. ":80" shows the web processes.
     3. ":8888" shows the NetCat process.
     4. Replace **1012|1972** with your process IDs (Pid), and separate each Pid with a pipe(**|**).



1. View and Interrogate Network Connections
   * **Instructions**
     1. **./vol.py --profile=WinXPSP3x86 connscan -f /var/forensics/images/WV01\_clean.dd**
     2. **for** **IP** in `**./vol.py --profile=WinXPSP3x86 connscan -f /var/forensics/images/WV01\_clean.dd** | **grep ":80"** | **awk '{print $3}'** | **sed 's/:80//'** |**uniq**`; **do** **WHO=`whois $IP | grep "Organization:"`**; **echo "IP: $IP | $WHO"**; **done**
     3. Record one of the Process IDs (PID) associated with web traffic on port 80
   * **Note(FYI):**
     1. **./vol.py --profile=WinXPSP3x86 connscan -f /var/forensics/images/WV01\_clean.dd,**this command looks at all TCP connection both **open and terminated.**
     2. **for, do, and done,**is the skeleton of a for loop in Bourne (sh) shell.
     3. **grep ":80"**, filter or display only port 80 connections of the vol.py results.
     4. **awk '{print $3}'**, cut out the third column of the vol.py results.
     5. **sed 's/:80//'**, remove the string ":80" from each IP Address.
     6. **uniq**, remove any duplicate IP Addresses.
     7. **WHO=`whois $IP | grep "Organization:"`**,do a whois lookup on each IP address and assign the results to the WHO variable.
     8. **echo "IP: $IP | $WHO"**, print out the IP address and organization to the screen.



1. View DLL used by a Running Processes
   * **Instructions:**
     1. **./vol.py --profile=WinXPSP3x86 dlllist -p 1012 -f /var/forensics/images/WV01\_clean.dd**
   * **Note(FYI):**
     1. PID **1012** is my process ID associated with Internet Explorer.  Supply the PID you recorded in the previous step above.
     2. The purpose of this tool is to list all the dll's that are associated to the particular process ID, which happens to be Internet Explorer.



**Section 5: Proof of Lab**

1. Proof of Lab
   * **Instructions:**
     1. mkdir -p /var/forensics/images/WV01\_clean
     2. ./vol.py imageinfo -f /var/forensics/images/WV01\_clean.dd > /var/forensics/images/WV01\_clean/imageinfo.txt
     3. ./vol.py --profile=WinXPSP3x86 pslist -f /var/forensics/images/WV01\_clean.dd > /var/forensics/images/WV01\_clean/pslist.txt
     4. ./vol.py --profile=WinXPSP3x86 connections -f /var/forensics/images/WV01\_clean.dd > /var/forensics/images/WV01\_clean/connections.txt
     5. ls -l /var/forensics/images/WV01\_clean/\*
     6. date
     7. echo "Your Name"
        + Note:
          1. Replace the string "Your Name" with your actual name.
          2. E.g., echo "John Gray"
   * **Proof Of Lab Instructions:**
     1. Press the <Ctrl> and <Alt> key at the same time.
     2. Press the <PrtScn> key.
     3. Paste into a word document
     4. Upload to Moodle

