

Memory Analysis Project Setup

Chris Neilson

August 19, 2009

1 What This Project Entails

2 What You Need

This list is to get you started, feel free to add or remove anything you would like.

- Copies of Windows XP Service Pack 2 on virtual machines or physical machines.
- Some way to get files to the machines, it can be tricky with virtual machines.
- HackerDefender 1.0, a well known, freely available, rootkit.
- netcat.exe for Windows, similar functionality to netcat for Linux.

3 What We Used

- Windows XP Service Pack 2, no security updates, on virtual machines using Qemu.
 - <http://www.qemu.org/>
- HackerDefender 1.0 as rootkit.
 - http://www.rootkit.com/board_project_fused.php?did=proj5
 - From our server http://our_server
- netcat.exe for Windows
 - <ftp://ftp.cosoft.org.ch/incoming/nc.exe>
 - From our server http://our_server

4 Setting Up the Infected Image

This section covers the steps to setup images like we did. For more in depth instructions on Qemu....

4.1 Setting Up Windows Users

1. Follow this guide from Microsoft:
<http://www.microsoft.com/windowsxp/using/setup/winxp/accounts.mspx>

4.2 Setting Up netcat.exe

The best solution for this is to make it run at startup using HackerDefender however I have been unable to get this to work. So what I did was try to make it appear as if it started up when the computer booted using the following steps, the only issue is that the timestamps will be off.

1. Put the command shown below for the listener in the startup section of the HackerDefender.ini
2. Open a DOS shell and navigate to the location of netcat.exe
3. Run the command `netcat.exe -L -d -p 9999 -t -e cmd.exe`
4. Since the -d switch is given, you may close the shell after you start the process and it will run in the background.

4.2.1 What does this do?

This will run netcat to listen on the port 9999 and when it has someone using netcat connect to port 9999, it will execute a command shell which will be usable by the person connecting to port 9999. If someone wanted to do this all they would need to know is the IP of the machine running the listener and then executing `netcat.exe <listener IP> 9999` and they will have a shell on the listening machine.

4.2.2 What this will be used for in the project

This listener will create a suspicious process that can be found with the use of the `pslist` and `sockets` tools of Volatility. With further investigation into the HackerDefender memory, it is possible to determine that this process was launched as a startup process of HackerDefender.

4.3 Setting Up HackerDefender

IMPORTANT: When setting up the HackerDefender .ini file, make sure to keep track of what the Service-Name is set to. The default is usually HackerDefender100. Knowing the service name allows you to easily stop HackerDefender and reconfigure it. To stop HackerDefender, see the instructions below.

Customize the HackerDefender .ini file to your liking. It is pretty powerful and very flexible. Reading the basics of the readme is a big help although its fairly self explanatory. Hide files and processes, such as `RootkitRevealer*`. Add the netcat.exe command from above to the Startup section. Make sure the HackerDefender .exe and .ini have the same name and are in the same location. It is also sensible to hide these names and locations. Once you have customized the file to your likings follow these simple steps.

1. Double click the .exe
2. Wait a few seconds, refresh the folder, the files should be gone
3. Reboot the computer to finalize the install and remove any artifacts from memory that may be present.

4.3.1 What does this do?

This installs and runs a rootkit. This rootkit is mainly a user-mode rootkit that hooks Windows API calls and modifies the outputs. It allows files and processes to be hidden as well as many other things. For a more in-depth discussion on rootkits visit <http://en.wikipedia.org/wiki/Rootkit>. Hiding the file `RootkitRevealer*` will make it so if you try to download and run RootkitRevealer.exe from *Windows Sysinternals*, it simply won't run.

4.3.2 What this will be used for in the project

The rootkit is the most important part of the project. Rootkits can totally hide malware from virus scanners and be used to setup gigantic botnets without the users of the computers knowing. The most fail proof way to detect these rootkits is through memory analysis. Volatility shows these hidden things as clear as day, in fact it may be hard to determine if it was even hidden on the host computer. The majority of the questions for this project are created from the way HackerDefender works. The students should be able to determine some of the API calls that are hooked and much more.

4.3.3 How to stop HackerDefender

1. Open a DOS shell.
2. Run the command `net stop HackerDefender100` Where the last argument is the exact service name as it appears in the .ini file. This may take a few minutes and may give an error, however if you got the service name correct, this is made to stop HackerDefender. Processes and files will be unhidden and things will go back to normal. Although all HackerDefender files must be removed to totally cleanse a system.

4.4 Finalizing The Setup

4.4.1 Other Programs

Run some other programs to make the computer appear in use. Some of these programs could be web browsers, solitaire, word processors, or anything else that is usually found on a computer

4.4.2 When to take memory images

You will want to take two memory images of the infected machine. The first memory image should be running for awhile and have many programs open. The second should be right after a reboot. If you started netcat.exe manually, make sure to start it again. Take this image with nothing else running, at least that you started. The clean image should be a totally clean install of Windows XP SP2 that can be used as reference while using Volatility.

5 Taking Memory Images

5.1 How to take a memory image using Qemu

In this instance our Qemu image is running off of 512 megabytes of ram.

1. While running Qemu press control-alt 2, this will bring up a sort of Qemu shell.
2. Use the command `pmemsave 0 536870912 memoryimage.dump`
 - The format of this command is `pmemsave <start offset> <end offset> <output name>` where the offsets must be in bytes.

5.2 How to take a memory image without Qemu

There are obviously more than one ways to do this but this works well.

1. Download an open source memory imaging program such as win32dd.exe
 - <http://win32dd.msuiiche.net/>
2. Open a DOS shell and navigate to where win32dd.exe is located.

3. Use the command `win32dd.exe -r memoryimage.dump`. Note: this process can take several minutes.