Rootkit Investigation Documentation
Date Started: 3-30-2011

Base Investigation Environment:

- Running VMware Server 1.10
- Took a snapshot of the initial image to make live analysis easier
- Allowed the VMware image to have an active internet connection, since it was easier to load tools then. I would not do that during an actual investigation
- Used a Backtrack Linux v4 live boot cd, and WindowsPE to help with offline analysis of the windows image.

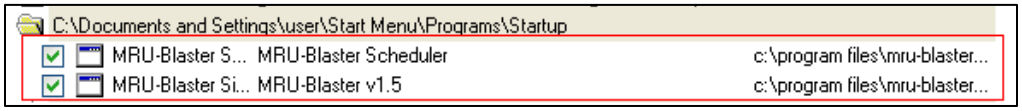
Tools:

- Windows SysInternal Tools: Provided an easy way to quickly scan the system and detect low-hanging fruit. For example, autoruns, processexplorer, rootkitrevealer, etc.
- WindowsPE: Initially used this in an attempt to run autoruns on an offline image. Ended up using this to examine files offline as well.

Investigation:

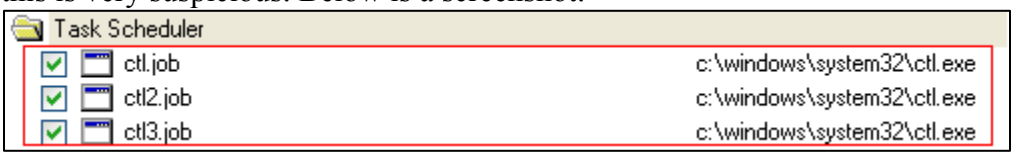
- 3-30-2011:
    - Started out running the SysInternals tool 'Autoruns' to see what showed up. It would be more effective to run it from a WindowsPE boot image against an offline system, but I was too lazy to create a WindowsPE image which contained that tool. Note, I am very aware that since this tool is running on the target system, the results may not be accurate.
        - For options, I set it to 'verify code signatures', and 'hide windows and Microsoft signed code'.
        - The first suspicious file was fat.vbs which is set to run the file fat.bat at startup. It was misleadingly named, and did not have a valid code signature. Below is a screenshot:

        - There were also two executables in the startup folder for mru-blaster. These might be legitimate, (since that also is the name of a legitimate program), but without a valid code signature there isn't enough information to make a determination at this time. Below is a screenshot:

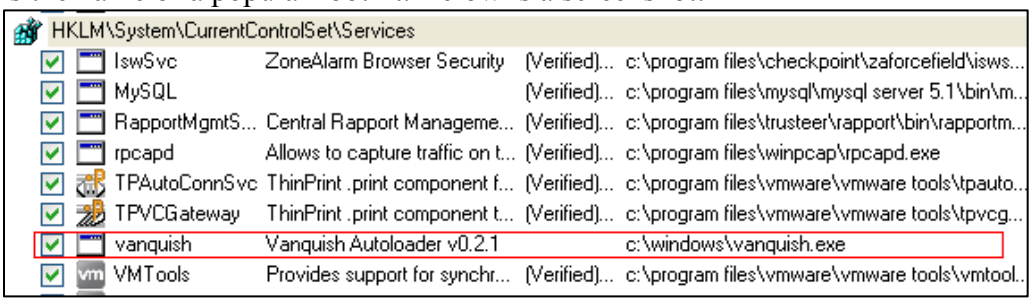


        - Task scheduler had three different jobs, but they all pointed to the same executable, 'ctl.exe'. Considering the location and no valid code signature,

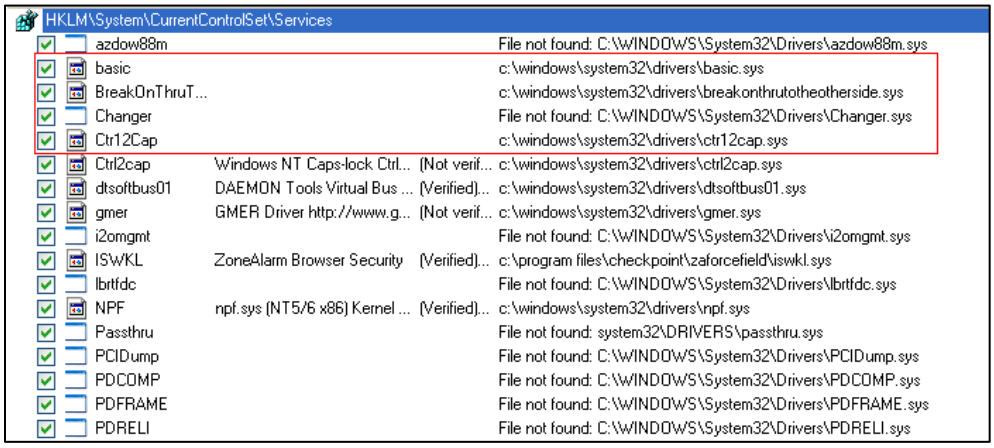
this is very suspicious. Below is a screenshot:



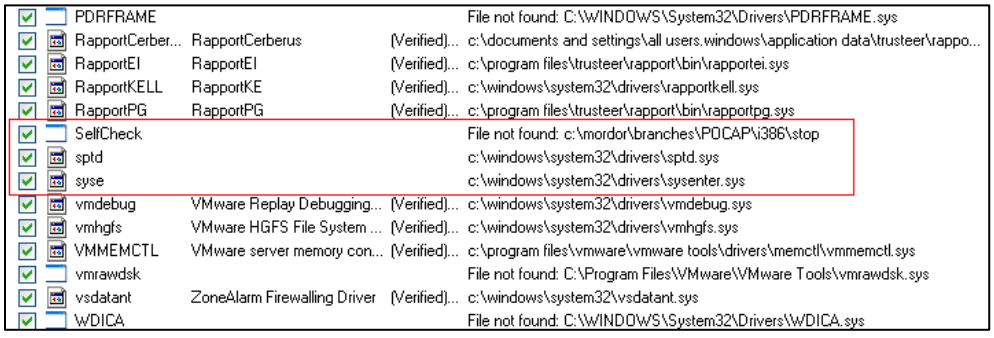
- The vanquish service was detected. This is very suspicious since vanquish is the name of a popular rootkit. Below is a screenshot:



- There were a large number of drivers that had bad links, or did not have valid code signatures. That was a bit suspicious. Many of the not-found are normal, like PCIDump, but the unsigned drivers could be anything. Below is the first screenshot:



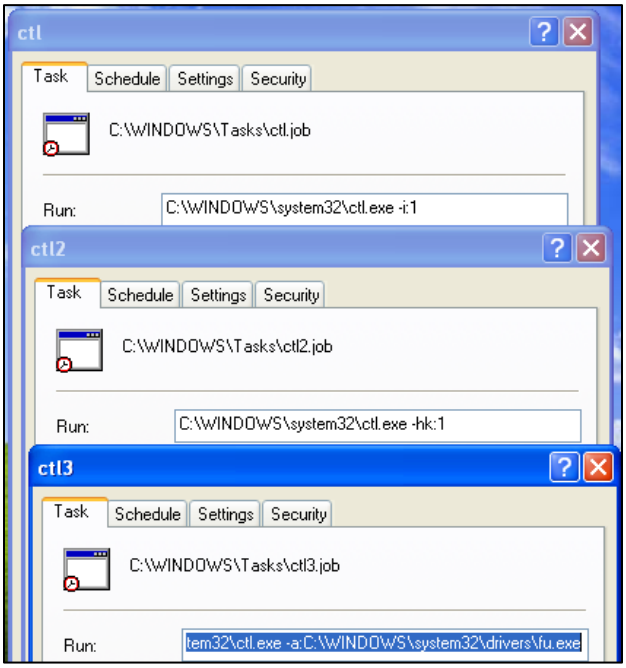
And here is the second screenshot:



- The vsockets library, “vsocklib.dll” was also detected as unsigned code, but that is a normal part of vmware tools, so it probably is not malicious.

- Next I ran the SysInternals tool ProcessExplorer. Once again, this was to help identify any low hanging fruit.

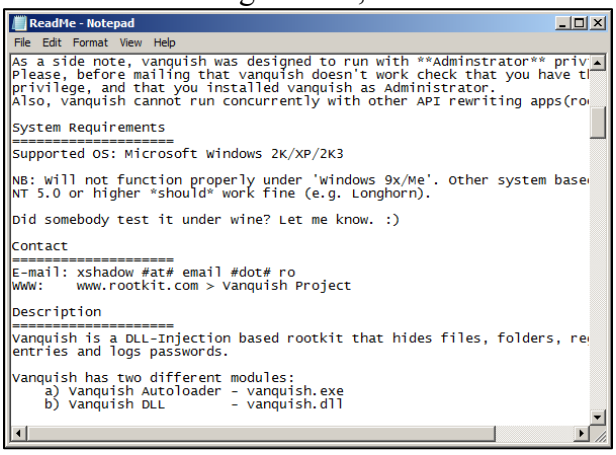
        - The main check I did was to see if any of the running processes were using suspicious .dll files. I defined suspicious as not having a company name of 'Microsoft Corporation', or the name of the company who write the executable. From that I saw that vanquish.dll was being used both by several of the svchost.exe processes as well as the taskmgr.exe process. This implies that even if the rootkit is not the vanquish rootkit, (despite the same name) it is a rootkit and it is being run, since a legitimate program should not be hooking those processes.
        - What was surprising was that ctl.exe was not observed as running.
    - Checking the 'Scheduled Tasks Folder'
        - Since autoruns reported that there were three scheduled tasks to run ctl.exe, the next step I took was to investigate the actual scheduled tasks to see what they were. All three tasks were set to run 'At Login'. They each had different command line options. They were:
            1. C:\WINDOWS\system32\ctl.exe –i:1
            2. C:\WINDOWS\system32\ctl.exe –hk:1
            3. C:\WINDOWS\system32\ctl.exe –a C:\WINDOWS\system32\drivers.fu.exe
        - A screenshot of the tasks can be seen below:



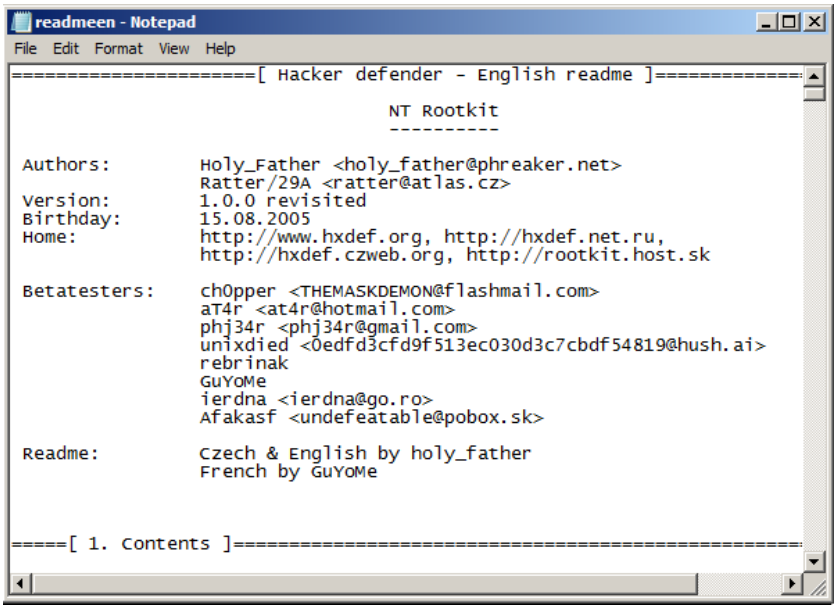
        - A Google search for ctl.exe brought up a lot of results mentioning it is malware, though I didn't see any results that talked about it in detail. FU is the name of a known rootkit, so chances are these are not legitimate files.
- 3-31-2011
    - Since I had such success running autoruns on the target machine, I wanted to see what would happen if I used it when the target machine was offline. To do this, I created a version of WinPE that contained the autoruns tool, and then booted into it. Note, WinPE loads the target disk as writeable by default, which is something to keep in mind during analysis. Unfortunately I was unable to get autoruns to

work in an offline setting, but since I had already loaded WinPE I decided to look around to see if anything stood out.

- Browsing the files in WinPE, I saw the directory C:\vanquish-0.2.1. Opening it I found the following readme, which shows that it is in fact the vanquish rootkit.

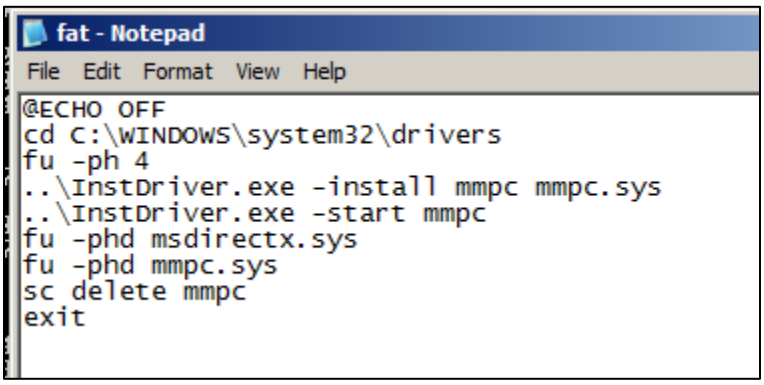


- There was a second directory, c:\hxdef100r, which contained the readme and several executables for the hacker defender rootkit. The readme is pictured below:

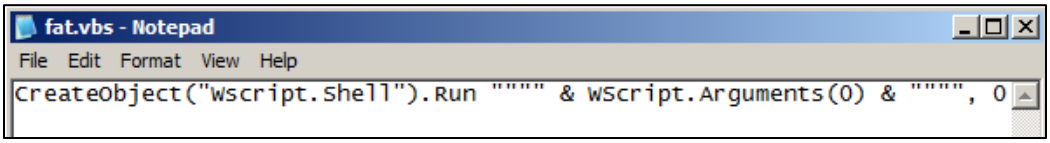


- Also of note, according to the time-stamps, vanquish was installed on 3/06/2011, and hackerdefender was installed on 3/13/2011. Due to this, I started looking for files that were modified in the last month. This was made easier by the fact that this is a demo VM machine.
- One file that stands out is fat.bat found in the c:\WINDOWS\ directory. It loads and runs the driver mmpc.sys and then hides it along with msdirectx.sys using the

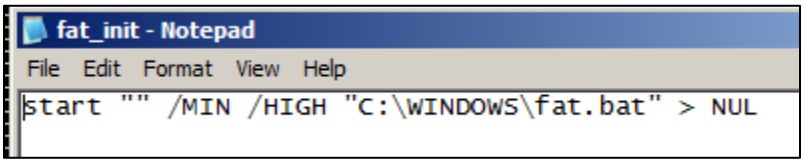
tool ‘FU’. A screenshot of that can be seen below:



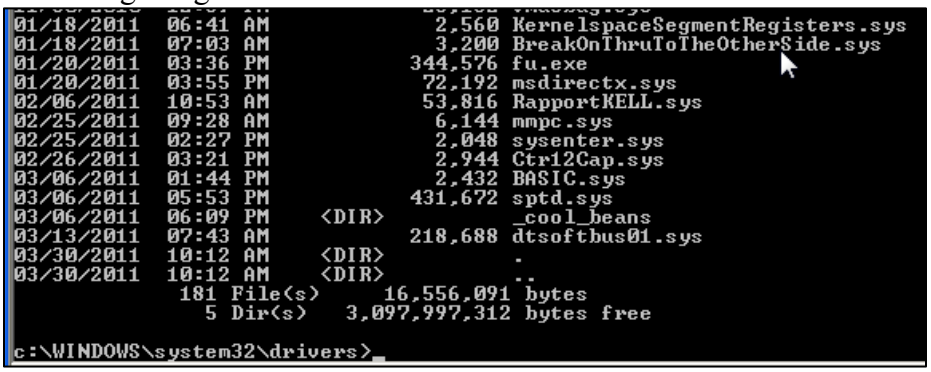
- Looking at the fat.vbs program that I noticed earlier in autoruns, I saw that it simply runs the command that is given to it via its startup script. A screenshot of it is below:



- Furthermore, I noticed fat_init.bat which invokes fat.bat. A screenshot of it is below:

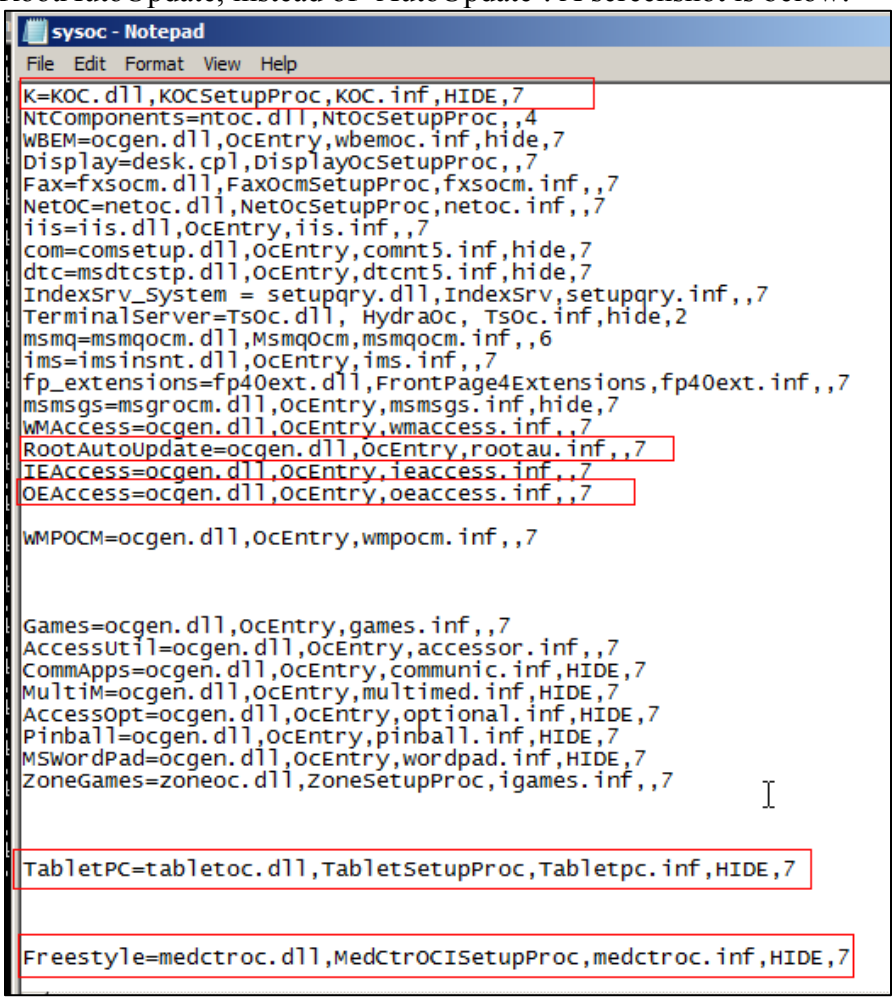


-
- Looking at the c:/WINDOWS/system32/drivers directory showed some interesting things:



- Syscoc.inf also contained some suspicious entries, especially since they were set to “HIDE”. Examples of that are k=KOC.dll, and Freestyle=mdctroc.dll. Also, some of the entries had different names than expected. For example

RootAutoUpdate, instead of 'AutoUpdate'. A screenshot is below:



- Looking into the corresponding inf files, rootautoupdate, and OEAccess appear to be legitimate. Also KOC.dll appears to be a part of winrar.
- Further looking into this, the entries actually look legitimate, and this was a false lead.

- April 1st, 2011:
  - The next thing to check was those suspicious drivers I saw in autoruns. Unfortunately, I forgot to load 'strings' onto the WindowsPE ISO, but I was able to view them in notepad to see if anything showed up.
  - The file 'basic.sys' had the following string in it which highly implied it was one of the rootkit files:
    - **Z:\binarytransfer\rootkits___installed\basic_callgate\callgate_driver\objfre_xwp_x86\BASIC.pdb**
  - Judging by the name, and strings, from BreakOnThrouToTheOtherSide.sys, it appears to be the same file used in the Intermediatex86 class. It displays processor sessings like the interrupt flags, esi, edx, etc. If I had more time I might want to confirm it, (via a hash), to make sure it is the same file, but for now it looks like a legitimate driver.
  - The next driver investigated was Ctr12Cap.sys. Considering the name is similar to a legitimate Sysinternals program, (Ctrl2Cap.sys), which was also on the

computer, but different, (swap the 'l' with a 'one'), and it was compiled on c:\bhf\objfre_wxp_x86\i386\Ctr12cap.pdb, it probably is a 'bad' file. Looking at some of the calls it makes such as 'KeBugCheckEx', 'KeTickCount', 'KeServiceDescriptorTable', etc, it looks like it has the ability to hook a process and has the possibility to reboot the system.

- Sptd.sys is 'weird' since it is a legitimate file used in VMWare, in the code it looks like it has a signature, 'VeriSign Class 3 code signing 2009-2', but in autoruns it does not have a valid signature. I'll make it as suspicious, but ideally I'd like to have a valid copy of the file to compare against it, since there is a good chance it is legitimate.
- Sysenter.sys: The string 'ROOTKIT: OnUnload called' was seen in the code. Also the path 'C:\sysenterhook\objchk_wxp_x86\i386\sysenter.pdb" was seen. I'm going to go out on a limb and assume this is a rootkit example that hooks sysenter ;)

- April 3rd 2011:
  - Well, it's time to actually try and fix this system. If this was a real life case my advice would be to take off and nuke it from orbit – It's the only way to be sure ;) Aka rebuild the machine from scratch. It's been thoroughly and I probably haven't detected all of the 'bad' software on it. Since this is a class though, I'll try and remove what I can and see if that breaks anything.
  - I double-checked my backups in case I deleted anything important and broke the windows install. In real life this is why DD copies are nice since you would want to save a copy of the malicious files for future analysis if needed.
  - The first thing to do is remove the known malicious code. The following files and directories were deleted:
    - The following files were in c:\Windows\system32\drivers
      1. Sysenter.sys
      2. Ctr12Cap.sys
      3. Basic.sys
      4. The _cool_beans directory was hidden, but there was nothing in it which was weird. It's not malicious as far as I can tell, but it was being hidden by malicious programs so I deleted it to be on the safe side.
      5. Fu.exe
    - The following files were in C:\Windows\
      1. fat.bat
      2. fat.vbs
      3. fat_init.bat
      4. vanquish.exe
      5. vanquish.dll
      6. Hey did you know the gmer anti-rootkit was on here? Neither did I until now. Wait, looking at my notes, I did see it before in autoruns…
    - The following files were in C:\Windows\system32\
      1. He4HookInv.sys – Got lucky and saw this one when eyeballing the dir results. Noticed it's install date was 6:35 PM 11/02/2002

2. Ctl.exe
   - The following files were in C:\
     1. Deleted the entire vanquish d-0.2.1 directory along with vanquish.log
     2. Tried to delete the hxdef100r directory but it said access was denied. Changed the write protection using attrib –r, and was able to delete the directory.
   - The following files were in C:\Windows\tasks\
     1. Ctl1.job
     2. Ctl2.job
     3. Ctl3.job
- After double-checking my notes to make sure I didn't forget anything that I had already found, I rebooted the system to see what would happen.
  - It booted up ok, though of course I received an error for ' Can not find script file "C:\Windows\fat.vbs". Didn't know how to disable that from an offline copy of Windows.
  - Ran autoruns again to disable it from attempting to load deleted files, (and also see if anything new popped up).
    1. Heh, a 'missing file for HackerDefender showed up. Its company name was "powerful NT rootkit" ;)
    2. Ctrl2cap (the sysinternals one), showed up. It didn't have a valid signature though, which was … weird
    3. The gmer driver also showed up.
    4. Everything looks ok
- Checked the browser extensions for IE. Found a weird one labeled {C95FE080-8F5d……} It didn't have any files or folders associated with it so I disabled it.
  - I also disabled Diagnose Connection Problems, and Windows Messenger since they did not have a publisher associated with them, and I doubt that they would be missed by a user if they are legitamite
  - I also checked the search providers and accelerators, but nothing suspicious was found there.
- Tried to run the gmer anti-rootkit tool. That turned out to be mostly useless since it detected so many legitimate files. Nothing malicious stood out. It probably would be very useful if you had a 'known good' scan to compare it to, but since I didn't, this didn't provide much help.
- Ran SysInternals rootkit revealer. Nothing looked out of the ordinary.
- I wouldn't trust this system, but for the purposes of the class I'm going to declare victory, and submit my report.