

Answers to the Malware Analysis Part 2 questions.  
Tyler Hudak

1) Is this file Packed? If so, which packer was used?

Yes - the file is packed with UPX and is a self-extracting RAR archive.

2) Which command did you use to identify it?

I used the file command to find this out. The particular file I used was within Cygwin.

```
$ file malware-quiz.exe
malware-quiz.exe: PE executable for MS Windows (GUI) Intel 80386 32-bit, UPX
compressed, RAR self-extracting archive
```

3) Do you believe that is there any other way to identify the packer?

Yes. You could run it through a strings program to look for any readable strings. I typically use BinText for this, but the UNIX or sysinternals 'strings' command would work just as well. Just make sure you use the "-a -n 3" options to pick up any readable strings of length 3 or greater.

When you do so, you will see the UPX0, UPX1 and UPX! strings indicating it is packed with UPX. You will also see some strings referencing WinRAR and RAR indicating it is compressed with WinRAR.

You could also use a packer identification program such as PEiD (<http://peid.has.it/>) which will take a PE executable and analyze it for any known packers. However, you have to be very careful with this as it will sometimes actually load the executable in memory and run it to find which packer is used, which could lead to unintended execution of the malware. Not a good thing.

3a) Please describe the directory which this file will be installed?

The file will be installed into c:\windows\temp. This can be found by unpacking the file (upx -d -o unpacked.exe malware-quiz.exe), opening it in an RAR extractor like WinRAR and looking at the RAR SFX script. The PATH SFX archive command is where the file will be installed.

You could also have found this out by actually executing the file and watching it with filemon, regshot or a similar tool.

4) In the process to unpack this file, please describe all the options that you saw. And by 'describe' I mean tell me what does it do when unpacking or not...

The SFX archive comment for the RAR archive is shown below. The SFX archive comment is a list of commands which gives instructions to the RAR self-extractor on how it should extract the files. Next to each command (in bold) is a description of what it does.

**Path=c:\windows\temp** – This tells the self-extractor where to save the file. In this case, it is c:\windows\temp.

**SavePath** – This instructs the extractor to save the path of the archive to the registry and then restore it when executing the archive later.

**Silent=1** – This extracts the archive without displaying and dialog. Since the option here is 1 everything is hidden from the user. If the option was 2, the confirmation would not be shown to the user but they would see the progress indicator.

**Overwrite=1** – This tells the self-extractor to overwrite all files without confirmation from the user.

**Title=.^^0wn3d^^.** – This will set the title of the self-extractor window. However, since this is a silent install, the user will never see it.

#### **Text**

{

**just kidding...**

} – The self-extractor will add the text specified within the curly braces to the dialog window shown to the user upon self-extraction. Again, since this malware is a silent install, the user will never see it. It should be noted that the text within can be HTML or plain text.

#### **License=Malware License**

{

**This malware was created for educational purpose only.**

**Check [handlers.dshield.org/pbueno](http://handlers.dshield.org/pbueno)**

} – This will display the text within the curly braces to the user as a software license and the user will be given the option to accept it to continue extraction or decline it and stop extraction. Once more, however, since this is a silent install the user will never see this. This can be HTML or plain text as well.

#### 5) What does this malware do?

The malware is a command line executable, which when run, will display “Oh my...am I a really malware????” on the screen 99 times. In order to fully see this the malware has to run in a command shell. If you don’t do this, the output will quickly be displayed to the screen and then disappear. The malware does not open or write to any unusual or unnecessary files or registry keys. It also does not open any ports or send out any network traffic.

I found this out by first running the unpacked, uncompressed executable through BinText, a Windows strings program. There were a number of readable strings in there, two of which were interesting:

Oh my...am I a really malware????

and

Do you know what is the meaning of life? It is 42!

The first string is what we will see in the output. The second is the answer to the bonus question. As a side note, the strings output also shows that the malware was compiled with the Open Watcom C/C++ compiler. There is also the string “!this is a Windows NT character-mode executable” which indicates that it should probably be run at a command shell in order to see everything.

Next, I uploaded the malware to a Windows XP VMWare session set with Host-only networking to the Host OS (which was Mandriva). I opened up a command shell and ran a 1<sup>st</sup> shot of regshot. Regshot is a utility which, when run after the malware is run, will look for any added, modified or deleted files or registry keys. I also started regmon, filemon and tdimon to monitor the system for any registry, file or networking accesses during the malware execution. Finally, I started a sniffer on the VMWare networking port on the Host OS to look for any network traffic.

After running the malware from the command shell and seeing “Oh my...am I a really malware????” display in the window 99 times I stopped the \*mon utilities and the sniffer and ran the 2<sup>nd</sup> shot of regshot. The regshot compare and the output of the \*mon utilities and sniffer showed that the executable had not done anything else to the system.

It probably would have been just as easy to open the executable up in a disassembler/debugger like IDAPro, but I did not have it on the system I was analyzing the malware on.

6) And finally, as a bonus question: What is the meaning of life?

42. This can be found by looking at the readable strings in the uncompressed and unpacked malware executable or it should be known by Douglas Adams fans everywhere. ☺