

Malware Analysis Quiz 7

Write up by: Zach Jansen

zach.jansen@gmail.com

Analyzing this piece of malware was indeed more challenging than usual as I will outline below. The quizmal malware attempts to protect itself from analysis using the Themida¹ application packer. I was not able to analyze it to my satisfaction, although with a little more time I'm sure more interesting things would have turned up. I will first describe what steps I went through to analyze the malware, and then at the end summarize my answers to the 7 quiz questions posed regarding the malware.

Step one for analyzing the malware is of course to obtain it. I downloaded quizmal.zip using a virtual machine. Initially I started using Microsoft's Virtual PC, however, as I soon found out, this piece of malware does not enjoy running inside of a vm environment. More on that later. I also verified the md5 sum using the md5summer tool, and matched the md5sum with what was posted on the malware quiz page.

The next stage in the analysis was to figure out what kind of malware this might be using a bit of static analysis. Running Foundstone's BinText program to get a list of text strings from the executable yielded very few legible results, suggesting that the executable was packed. This isn't really a surprise as most malware I see these days is packed with something. Of the few legible strings, one was "Themida" and it was a good guess that quizmal3.exe was packed with something called Themida. Output from PeiD confirmed that it was indeed packed with Themida version 1.0.0.5. Themida is a commercial packer, apparently popular in China, which supports many advanced features such as anti-debugging and virtual machine detection. Newer versions claim to support anti-dumping as well to prevent another program from dumping the unpacked executable from memory while the program is running. There is a good post² on Themida on the ISC web site by Lenny Zeltzer explaining some of Themida's capabilities and how to get around them. Amusingly enough the post happened during the malware challenge giving all of us a good hint about how to get around the anti-vm capabilities if we hadn't discovered it already.

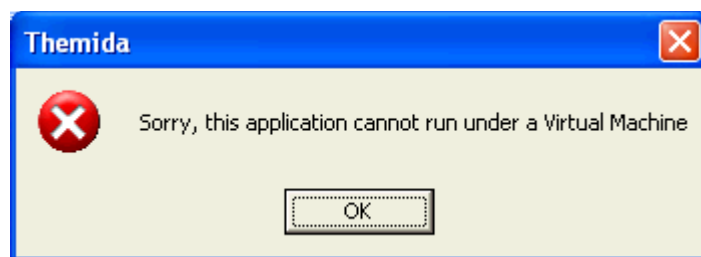
Since getting a listing of the strings in an executable is fairly important for understanding what it does, getting quizmal3.exe unpacked is fairly important. Without a strings listing some things would be missed, for example a secret message embedded within the executable =) Generally speaking there are two methods that I normally use to unpack an executable. First is to find a program that unpacks it for you. Common packers such as UPX have readily available unpackers that can be downloaded from various web sites such as www.exetools.com or are simply built in to the packer itself. The other option is to use a debugger such as Ollydbg and actually run the executable until it unpacks itself. While researching, I found a third option, a "new to me" tool called Sysanalyzer that would also dump an unpacked version presumably from the running malware's memory space.

Searching for an unpacker yielded a program called XprotStripper³ that would dump the program and let me get a strings dump. I had some trouble using it in Virtual PC since quizmal3.exe won't run in a virtual machine environment because the Themida packer is detecting Vmware and then exiting the program. It does give a fairly helpful error message though:

1 <http://www.oreans.com/themida.php>

2 <http://isc.sans.org/diary.php?storyid=1871&rss>

3 <http://forum.exetools.com/showthread.php?t=6527>



It is very polite, simply informing me that it didn't want to run under a virtual machine. I needed to find some way to make Themida believe it wasn't running inside of a virtual machine environment, aka, the Matrix. A little bit of searching led me to a paper called “Thwarting VM Detection”⁴ by Ed Skoudis and Tom Liston who pointed out that there are some undocumented configuration options in VMware that would fool most vm detection mechanisms. The configuration options are:

```
isolation.tools.getPtrLocation.disable = "TRUE"
isolation.tools.setPtrLocation.disable = "TRUE"
isolation.tools.getVersion.disable = "TRUE"
isolation.tools.getVersion.disable = "TRUE"
monitor_control.disable_directexec = "TRUE"
monitor_control.disable_chksimd = "TRUE"
monitor_control.disable_ntreloc = "TRUE"
monitor_control.disable_selfmod = "TRUE"
monitor_control.disable_reloc = "TRUE"
monitor_control.disable_btinout = "TRUE"
monitor_control.disable_btmemspace = "TRUE"
monitor_control.disable_btpriv = "TRUE"
monitor_control.disable_btseg = "TRUE"
```

After adding these options to my virtual machine's vmx file I was able to get the malware to run and get the unpacked dump using the XproStripper program as mentioned above.

There are quite a few interesting strings in the executable, some of which I've included below with my comments:

1. PONG – a response message to an IRC PING message. PING – PONG, get it?
2. JOIN #secretcow werule
3. Daddy
4. C:\DI
5. PRIVMSG #yousawthatss :Downloading. - Sends a “Downloading” message to channel “yousawthatss”
6. PRIVMSG #yousawthatss :Executing.- Sends a “Executing” message to channel “yousawthatss”
7. uk.undernet.org – IRC server to connect to.
8. RestartApp.exe
9. #testbot – a channel name
10. ircbot

4 http://handlers.sans.org/tliston/ThwartingVMDetection_Liston_Skoudis.pdf

11. no security without knowledge. No knowledge without research. Be a good guy! It is worthwhile. - A secret message?

In summary what I see here are some IRC commands and channel names, with passwords where applicable. There are also a couple of executable references here that lead me to wonder if the incident response guys need to double check the user's machine to see if dl.exe or restartapp.exe exist there. I'll speculate a bit more on this later. There is also the “#testbot” and “ircbot” strings which are a bit out of place. As far as I could tell the executable never attempts to join the testbot channel, so I wonder if they were part of someone's attempt to write and test their own IRC bot that just ended up not getting used in the final product. A google search for those terms reveals a lot of hits for sample IRC bot code.

Now that I've pretty much exhausted the static analysis that I can do (unless I want to disassemble the code. I don't), it's time to move on to dynamic analysis of the executable. Here I began experimenting with a new tool called Sysanalyzer⁵. I found Sysanalyzer while looking through Lorna Hutchinson's post⁶ on malware tools of the trade. It's not referenced there, but another tool called the Malcode Analysts Pack from iDefense is and Sysanalyzer was right below it. Sysanalyzer will launch an executable and create a report giving you a general idea of what the malcode does. It will watch for new files and registry keys that are created, new executables, and open ports. One potentially interesting feature is it's ability to scan an executable and tell you what exploit it is using. Unfortunately that feature didn't work in this case and none of the 20 or so exploits it can detect were detected in quizmal3.exe. It will also get a strings dump from the running process, thus pretty much eliminating the need to go find an unpacker. At least in this case it did. I haven't played with it enough to know how well it works with other malware. One of the really handy features of Sysanalyzer is that it runs a sniffer and logs IRC traffic in a special window. This made it really easy to watch the malware in action as it joined a channel. Here is the abbreviated output of a session:

```
USER Hello2204025 Hello2204025@foo.bar Hello2204025 Hello2204025
```

```
PING :138650
```

```
PONG :138650
```

```
:London2.UK.EU.Undernet.Org 001 Hello2204025 :Welcome to the UnderNet IRC Network via Kewlio.net, Hello2204025
```

```
:London2.UK.EU.Undernet.Org 002 Hello2204025 :Your host is London2.UK.EU.Undernet.Org, running version u2.10.12.05
```

```
:London2.UK.EU.Undernet.Org 003 Hello2204025 :This server was created Mon Sep 4 2006 at 00:07:31 BST
```

```
:London2.UK.EU.Undernet.Org 376 Hello2204025 :End of /MOTD command.
```

```
:London2.UK.EU.Undernet.Org NOTICE Hello2204025 :on 1 ca 1(4) ft 10(10)
```

```
:Hello2204025!~Hello2204@(my IP address) MODE Hello2204025 :+i
```

```
JOIN #secretcow werule
```

```
:Hello2204025!~Hello2204@71.205.117.149 JOIN #secretcow
```

```
:London2.UK.EU.Undernet.Org 353 Hello2204025 = #secretcow :@Hello2204025
```

```
:London2.UK.EU.Undernet.Org 366 Hello2204025 #secretcow :End of /NAMES list.
```

```
PING :London2.UK.EU.Undernet.Org
```

⁵ http://labs.iddefense.com/software/malcode.php#more_sysanalyzer

⁶ <http://isc.sans.org/diary.php?storyid=1801>

PONG :London2.UK.EU.Undernet.Org

As we can see the malware joins the London2.UK.EU.Undernet.Org server, which has the same IP address as what the Incident Response team detected, 195.68.221.221. It also sets a NICK of Hello2204025. Hello appears to be hard coded, with the 7 digit number being random to prevent name collisions. After the welcome message from the IRC server (heavily abbreviated) it joins the “#secretcow” channel using the password “werule” and essentially waits, presumably for commands from someone within the channel. The PING and PONG messages are the IRC server checking to see if the malware is still connected. The server sends a ping, and if the client (malware in our case) is still connected it sends back a PONG message. I let the malware sit in the channel for a while but nothing happened. The lights are on but nobody's home? Perhaps the bot herder was sleeping or just wasn't sending instructions at the time. Or maybe there was no bot herder or it's not really malware. It's hard to say.

One thing I thought was interesting was that the malware activates a listening port and appeared to be listening for traffic. It did not listen on a set port, it always picked a new port each time it was run starting around 1024 and incrementing by 2 every time quizmal3.exe was rerun. The odd part was that I was completely unable to communicate with that port. I could not port scan it or connect to it with netcat. I will speculate that it was listening for traffic from a specific IP address, or perhaps as part of generally being contrary about being analyzed, this malware wouldn't response to IP addresses from private IP ranges. I never did figure out what exactly was listening on that port. A back door? Some kind of server?

Another interesting thing I noticed is that this malware does not make any attempt to rerun itself at startup. There were no registry keys or services created that would rerun it upon startup. The only service created was for the Themida packer itself and didn't rerun the malware. Additionally, I used Sysinternal's (or is it Microsoft's...) Rootkit Revealer and F-Secure's Blacklight to make sure it wasn't hiding a root kit along the way. Both scans came up blank. This makes me wonder if this malware is missing something, perhaps the dl.exe or restartapp.exe that were seen in the strings dump from the executable.

Last but not least, Themida will not run with Regmon or Filemon running. If quizmal3.exe is run while Regmon or Filemon are running it pops up a message asking you to close the analysis tools and rerun the application. How polite! I made a brief attempt to hide Regmon and Filemon using the Hacker Defender root kit, but Themida still detected it. Perhaps a kernel mode rootkit would have done a better job.

In summary of my analysis, this was one tough malware! Themida's ability to prevent analysis didn't prevent me from figuring out where it was connecting to, but it did make it hard to analyze and I'm still not sure what it's full capabilities are.

On to the quiz questions:

- 1) Is this malware packed? If so, with which packer?
 1. Yes, quizmal3.exe is packed using the commercial packer Themida⁷ version 1.0.0.5. See above for more details on Themida.

- 2) What is the purpose of this malware?

⁷ <http://www.oreans.com/themida.php>

This malware appears to be a custom IRC bot with partial IRC functionality. It appears to generate a random username (HelloXXXXXXXX where X appears to be a random numeral) and joins the #secretcow channel on the undernet IRC network with password “werule”. It also seems to report a message of “Downloading” or “Executing” to the channel “yousawthatss”. I would thus suggest that this is an IRC bot that works as a downloader. There are references in the exe to other executables such as dl.exe and restartapp.exe so I wonder if the incident response team needs to go back and look at this user's machine for said executables. It's possible that this program works in conjunction with those executables. What's odd is what this executable doesn't do. It doesn't make any attempt to set itself up to restart on reboot. That's an unusual bit of malware and another reason I think something is missing from this picture. It also opens a port, possibly as a backdoor.

3) Does it connect to a remote server? With which purpose?

Yes, it connects to London.uk.eu.undernet.org, aka 195.68.221.221, which is a standard IRC server, and part of the undernet IRC network.

4) Which channels does it connects to?

It connects to the “secretcow” channel. It also seems to send messages to the “yousawthatss” channel. It references the “testbot” channel as well, although I think that was part of testing this custom code and isn't part of the malware's evil plans to take over the world.

5) Can you get any passwords related to this malware?(Not the infected password) :)

It appears that a password is used to access the secretcow channel. Don't worry, you won't need to brute force this password with a supercomputer, it's “werule”.

6) Which capabilities does this malware have?

I wasn't able to test or observe this, but it appears to be an IRC downloader. It's capabilities would be downloading and executing additional malware. Probably for the purpose of making money from spyware installation.

Bonus question:

7) What is the hidden message? (if there is any...) :)

no security without knowledge. No knowledge without research. Be a good guy! It is worthwhile