Analyzing Unknown Malware

by R136a1 | Hobby malware analyst (Windows x86/x64)
@TheEnergyStory, @MalwareChannel | KernelMode.info

Friday, June 20, 2014

#9 Blitzanalysis: Embassy of Greece Beijing - Compromise

It's friday afternoon, I had a bit of free time and stumbled across this tweet by PhysicalDrive0 (thx!) two hours ago and thought to give it a try to finally add a new article to this Blog (first of 2014):

https://twitter.com/PhysicalDrive0/status/479921770838102017

So, I went to Google to search for the domain of the Embassy of Greece Beijing and added the (allegedly) malicious java file package that was found by PhysicalDrive0:

URL: http://www.grpressbeijing.com/1.jar (malicious!)

Next, I loaded the 1.jar file into Java Decompiler to get the source code. It showed, that the functionality is obfuscated in some way, e.g. the function *csfn(String paramString)* decrypts all strings by "removing" the numbers of the string parameter:

Figure 1: Simple string (de)obfuscation

csfn("64s33333e3333t333S55e666c777u5r333i534t76y2M34a55n76a88g666e44r2222") -> setSecurityManager

There are some other obfuscation techniques, but they are not important here. Instead, the following deobfuscated code line in the function init() gives us an idea where the actual payload is located:

Resp localResp = new Resp(csfn("234p34a55445c43654k632434234235")); -> pack

We can also see, that the java package contains a file named pack, so we open 7-Zip and unpack the file. A quick view with a PE viewer showed, that it is a x86 PE executable not even encrypted (*SHA256: b832e4b5a4829c8df6de7b42c5cb32ef25b5ab59072b4c2a7838404cd0dd5e5f*):

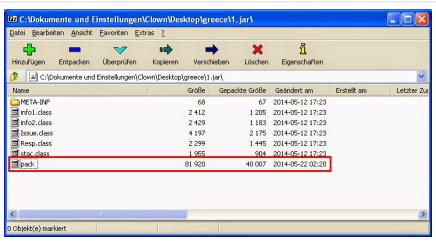

Figure 2: Payload inside Java package

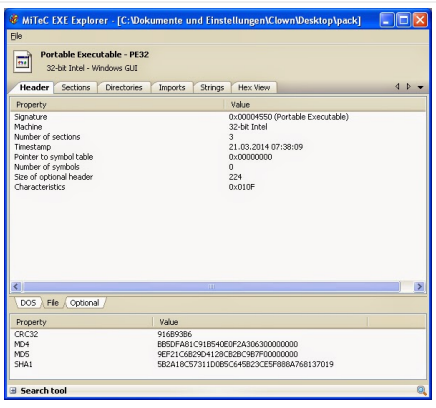

Figure 3: Payload inside PE viewer

So, I opened IDA Pro to take a quick look at the functionality. Together with the strings of the executable, we get a brief idea of what the purpose of this malware is. The important strings are as follows:

```
SELECT * FROM AntiVirusProduct
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings"  /v PrivDiscUiShown /t REG_DWORD /d  1  /f
reg add "HKCU\Software\Microsoft\Internet Explorer\Main"  /v DEPOff /t REG_DWORD /d  1  /f
reg add "HKCU\Software\Microsoft\Internet Explorer\Main"  /v DisableFirstRunCustomize /t REG_DWORD /d  2  /f
reg add "HKCU\Software\Microsoft\Internet Explorer\Main"  /v Check_Associations /t REG_SZ /d  no  /f
reg add "HKCU\Software\Microsoft\Internet Explorer\Main"
reg add "HKCU\Software\Microsoft\Internet Explorer\PhishingFilter"  /v ShownVerifyBalloon /t REG_DWORD /d  3  /f
reg add "HKCU\Software\Microsoft\Internet Explorer\PhishingFilter"  /v Enabled /t REG_DWORD /d  1  /f
reg add "HKCU\Software\Microsoft\Internet Explorer\PhishingFilter"
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings"  /v WarnOnPostRedirect /t REG_DWORD /d  0  /f
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings"  /v WarnonZoneCrossing /t REG_DWORD /d  0  /f
reg add "HKCU\Software\Microsoft\Internet Connection Wizard"  /v AutoRecover /t REG_DWORD /d  2  /f
reg add "HKCU\Software\Microsoft\Internet Connection Wizard"  /v Completed /t REG_BINARY /d  1  /f
\cmd.exe
```

Together with the output of IDA Pro, we can see that this malware uses the command line tool cmd.exe for adding several registry keys to Internet Explorer. also tries to retrieve possible AntiVirus information by using the COM interface (*dc12a687-737f-11cf-884d-00aa004b2e24 -> IWbemLocator -> SELECT * FROM AntiVirusProduct*). Furthermore, it makes use of the COM to launch an instance of Internet Explorer (*d30c1661-cdaf-11d0-8a3e-00c04fc9e26e -> IWebBrowser2*), supposedly to contact its C&C server. To verify this, we open up Wireshark and run the executable. As a result, we get the following networ information:

C&C server: defense.miraclecz.com (IP: 208.115.124.83)
HTTP GET request: /index.asp?id=50100

Also, we see that it downloads some kind of data (Base64 encoded). But first, we combine the C&C server and the HTTP request and open the URL in our favorite Browser:

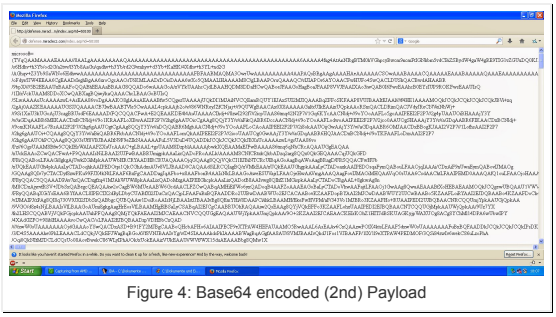

Figure 4: Base64 encoded (2nd) Payload

URL: defense.miraclecz.com/index.asp?id=50100

As you can see, there is a string named *microsoft* followed by Base64 encoded data. *Side note: Is there also a Linux equivalent?*

Next, we copy the Base64 encoded data and go to the following website to let us decode it into a file (because I had the feeling it's just another unencrypted executable):

http://www.motobit.com/util/base64-decoder-encoder.asp

As a result, we get another executable (*SHA256: a4863f44f48d1c4c050dd7baad767a86b348dd4d33924acf4e0a3cd40c6ae29f*) that was only Base64 encoded and not encrypted in any way:

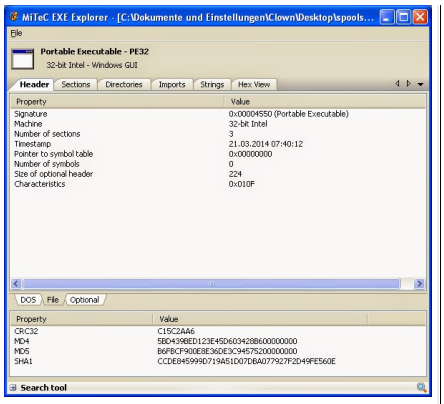

Figure 5: Downloaded Payload

So again, we fire up our PE viewer and take a look at the important strings:

```
http://buy.miraclecz.com
reg add "HKCU\Software\Microsoft\Internet Explorer\Main"  /v DEPOff /t REG_DWORD /d  1  /f
reg add "HKCU\Software\Microsoft\Internet Explorer\Main"  /v DisableFirstRunCustomize /t REG_DWORD /d  2  /f
reg add "HKCU\Software\Microsoft\Internet Explorer\Main"  /v Check_Associations /t REG_SZ /d  no  /f
reg add "HKCU\Software\Microsoft\Internet Explorer\Main"
reg add "HKCU\Software\Microsoft\Internet Explorer\PhishingFilter"  /v ShownVerifyBalloon /t REG_DWORD /d  3  /f
reg add "HKCU\Software\Microsoft\Internet Explorer\PhishingFilter"  /v Enabled /t REG_DWORD /d  1  /f
reg add "HKCU\Software\Microsoft\Internet Explorer\PhishingFilter"
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings"  /v WarnOnPostRedirect /t REG_DWORD /d  0  /f
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings"  /v WarnonZoneCrossing /t REG_DWORD /d  0  /f
reg add "HKCU\Software\Microsoft\Internet Connection Wizard"  /v AutoRecover /t REG_DWORD /d  2  /f
reg add "HKCU\Software\Microsoft\Internet Connection Wizard"  /v Completed /t REG_BINARY /d  1  /f
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Run"  /v spoolsv.exe /t REG_SZ /d  %%temp%%\spoolsv.exe  /f
spoolsv.exe
Software\Microsoft\Windows\CurrentVersion\Run
open file fail
cmd timeout error %d
Run cmd error %d
cmd.exe /c %s>%s
%s%d.txt
open file error
%temp%
%s%s.ini
myWObject
\cmd.exe
!DOCTYPE html
%s/?id1=blank%d&id2=%d%d
%s/?id1=%d%d
```

Again, we load the executable into IDA Pro and quickly fly over the assembly code to get an idea of the functionality. Once again, it creates several registry entries with the help of the command line tool and creates an instance of the Internet Explorer (*CoCreateInstance() -> d30c1661-cdaf-11d0-8a3e-00c04fc9e26e*) for contacting the C&C server. This time, the network information is as follows:

C&C server: buy.miraclecz.com (IP: 74.121.191.33)
URL parameters (from strings of executable):
%s/?id1=blank%d&id2=%d%d
%s/?id1=%d%d

From the code we can see, that the sample has also the ability to encode/decode data from/to Base64. The dynamic analysis showed the malware sample contacted the C&C server, but wasn't sending any URL parameters (id1, id2). Also the server didn't respond...

The files can be downloaded here: https://www.dropbox.com/s/ckr7p5kka62cc7s/Embassy%20of%20Greece%20-%20Beijing.zip
Password: "infected" (without "")

That's it, have a nice weekend...

Eingestellt von R136a1 um 9:38 AM

2 comments :

zeebrafin July 1, 2014 at 10:44 PM

Plz reupload the samples, the links are dead.

Reply

R136a1 July 2, 2014 at 1:04 AM

Thanks for pointing out! Curiously, the link was changed... (by Dropbox? oO)

Reply

Enter your comment...

Comment as: Google Account

Publish Preview

Newer Post Home Older P

Subscribe to: Post Comments (Atom)