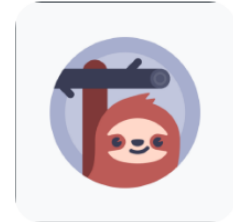# Conficker

github.com/itaymigdal/malware-analysis-writeups/blob/main/Conficker/Conficker.md

itaymigdal

itaymigdal/**malware-analysis-writeups**

Some of my Malware Analysis writeups.

| Malware Name	File Type	SHA256
Conficker	x32 dll	a30b63e1ed900d3f223277b1d3b09b045abc85600da0d3102fa61fb2bfc2ff99

Intro

Almost 15 years old ago, a worm named Conficker did a LOT of trouble. to this day, there are some Windows environments (mainly XP based networks) which are still infected with this piece of code (brilliant code for 2008). With millions of infections all over the world, 5 variations, and a lot of damage, some say this is the most remarkable worm that was ever made.
So i took it for a ride in my lab.

Analysis process

I first encountered that worm when i received a Disk On Key with an `autorun.inf` file and weird file with suspicious extension `jwgkvsq.vmx` which both were super infected in AV engines. Any time an infected DOK inserted into a computer, it pops up this window:
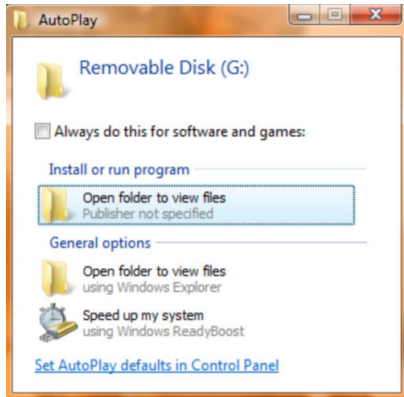
This is a very nice social engineering trick, the `autorun.inf` is disguised as the explorer icon and caption (look at the duplicated explorer actions, one under "Install or run a program" - which invoking the `autorun.inf` , and the other under "General options" - the benign one). Observing the autorun file:

We see a lot of shity unclear randomness which not clear if this is obfuscation or a binary. By scrolling down (how down? line 1227) some few strings are exposed inside this sea of garbage:

Cleaning it up:

```
icon = %syStEmrOot%\sySTEM32\sHELL32.Dll,4

shelLExECUte = RuNdLl132.EXE .\RECYCLER\S-5-3-42-2819952290-8240758988-879315005-3665\jwgkvsq.vmx,ahaezedrn

useAuTopLAY = 1
```

The first line bind the `autorun.inf` to explorer icon. The second line executes the other file using Rundll32.exe which invokes a gibberish export function (actually, this method isn't even exist in `jwgkvsq.vmx` dll. before validating the export name - DllMain is called).

Opening `jwgkvsq.vmx` in Pestudio:

property	value	value	value
name	UPX0	UPX1	UPX2
md5	n/a	89FFC709E081FAB549538498...	1F48A96EEFAE875C9C674C8...
entropy	n/a	7.787	3.663
file-ratio (52.76%)	n/a	52.45 %	0.31 %
raw-address	0x00000400	0x00000400	0x00015200
raw-size (86016 bytes)	0x00000000 (0 bytes)	0x00014E00 (85504 bytes)	0x00000200 (512 bytes)
virtual-address	0x10001000	0x10006000	0x1001B000
virtual-size (110592 bytes)	0x00005000 (20480 bytes)	0x00015000 (86016 bytes)	0x00001000 (4096 bytes)
entry-point	-	0x0001ABC0	-
writable	✗	✗	✗
executable	✗	✗	-
shareable	-	-	-
discardable	-	-	-
initialized-data	-	x	x
uninitialized-data	x	-	-
readable	x	x	x
self-modifying	✗	✗	-
blacklisted	✗	✗	✗
virtualized	✗	-	-

First stage is packed by UPX. unpacking:

```
PS C:\Windows\system32> upx.exe -d C:\Users\IEUser\Desktop\jwgkvsq.vmx -o C:\Users\IEUser\Desktop\confi_unpacked.dll
                    Ultimate Packer for eXecutables
                    Copyright (C) 1996 - 2020
UPX 3.96w       Markus Oberhumer, Laszlo Molnar & John Reiser    Jan 23rd 2020

        File size         Ratio      Format      Name
    --------------------   ------   -----------   -----------
     167640 <-    163032   97.25%   win32/pe      confi_unpacked.dll

Unpacked 1 file.
PS C:\Windows\system32>
```
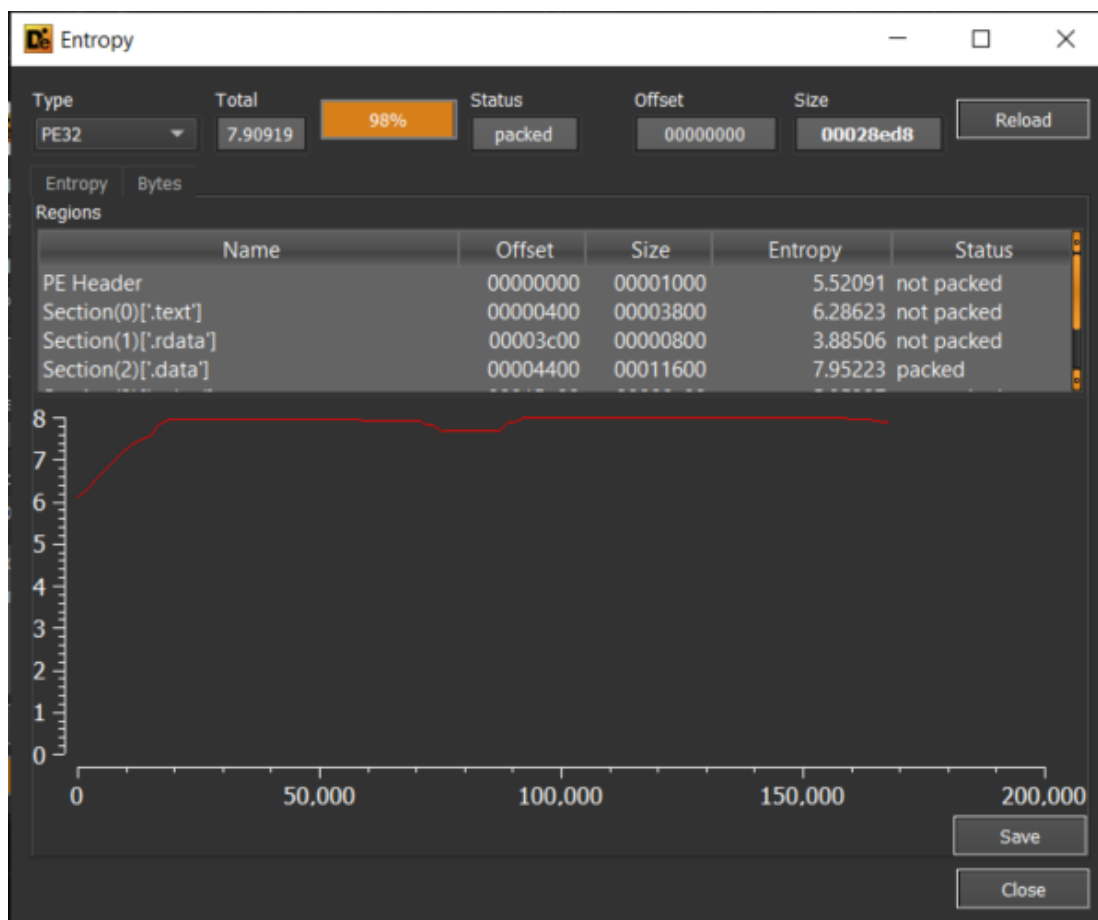
For my convenience, here i converted the dll to exe (the tool just changed a single bit in PE header):

```
PS C:\Users\IEUser\Desktop> .\dll_to_exe.exe
DLL to EXE converter v1.1
- for 32 & 64 bit DLLs -
args: <input_dll> <output_exe>
Press any key to continue . . .
PS C:\Users\IEUser\Desktop> .\dll_to_exe.exe .\confi_unpacked.dll confi.exe
[OK] Converted successfuly.
[OK] Module dumped to: confi.exe
PS C:\Users\IEUser\Desktop>
```

Entropy is 8 so the file is still packed:

We will try to unpack it later, for now let's run the file under Procmon to get a general idea of the file operations. The file is very noisy and many operation were seen.

The file persists itself in a run key:
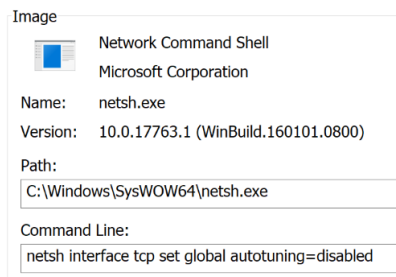
Date: 1/17/2021 4:21:58.2874657 PM
Thread: 9572
Class: Registry
Operation: RegSetValue
Result: SUCCESS
Path: HKCU\Software\Microsoft\Windows\CurrentVersion\Run\qbklwb
Duration: 0.0000997

Type: REG_SZ
Length: 132
Data: rundll32.exe "C:\Users\IEUser\AppData\Roaming\rhnhxm.dll",nejdswm

Deletes Windows Defender from run key:

Date: 1/17/2021 4:22:18.8731549 PM
Thread: 9572
Class: Registry
Operation: RegDeleteValue
Result: ACCESS DENIED
Path: HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Run\Windows Defender
Duration: 0.0000348

Resets the TCP receive window using Netsh.exe (not sure exactly why, but it's part of the setup for the upcoming Brute Force).

Image
    Network Command Shell
    Microsoft Corporation
Name: netsh.exe
Version: 10.0.17763.1 (WinBuild.160101.0800)
Path:
C:\Windows\SysWOW64\netsh.exe
Command Line:
netsh interface tcp set global autotuning=disabled

Probes for live hosts in the internal network by trying to connect to their SMB share:

4:22:55...	confi.exe	2360 TCP Reconnect	10.0.2.15:50669 -> 10.0.2.1:445
4:22:56...	confi.exe	2360 TCP Disconnect	10.0.2.15:50669 -> 10.0.2.1:445
4:22:56...	confi.exe	2360 TCP Connect	10.0.2.15:50674 -> 10.0.2.2:445
4:22:56...	confi.exe	2360 TCP Send	10.0.2.15:50674 -> 10.0.2.2:445
4:22:56...	confi.exe	2360 TCP Disconnect	10.0.2.15:50674 -> 10.0.2.2:445
4:22:56...	confi.exe	2360 TCP Connect	10.0.2.15:50675 -> 10.0.2.3:445
4:22:56...	confi.exe	2360 TCP Send	10.0.2.15:50675 -> 10.0.2.3:445
4:22:56...	confi.exe	2360 TCP Disconnect	10.0.2.15:50675 -> 10.0.2.3:445
4:22:56...	confi.exe	2360 TCP Connect	10.0.2.15:50676 -> 10.0.2.4:445
4:22:56...	confi.exe	2360 TCP Send	10.0.2.15:50676 -> 10.0.2.4:445
4:22:56...	confi.exe	2360 TCP Disconnect	10.0.2.15:50676 -> 10.0.2.4:445
4:23:00...	confi.exe	2360 TCP Reconnect	10.0.2.15:50677 -> 10.0.2.5:445
4:23:01...	confi.exe	2360 TCP Disconnect	10.0.2.15:50677 -> 10.0.2.5:445
4:23:04...	confi.exe	2360 TCP Reconnect	10.0.2.15:50678 -> 10.0.2.6:445
4:23:05...	confi.exe	2360 TCP Disconnect	10.0.2.15:50678 -> 10.0.2.6:445
4:23:08...	confi.exe	2360 TCP Reconnect	10.0.2.15:50679 -> 10.0.2.7:445
4:23:09...	confi.exe	2360 TCP Disconnect	10.0.2.15:50679 -> 10.0.2.7:445

In this part i started to debug the file under debugger in order to unpack it. Even though this is an old malware and fair to think that it is lacking protections, it's not true. it contains polymorphism, obfuscation and anti-analysis tricks. after some struggling with it and at least 5 `VirtualAlloc` , I saw a PE file that was written to a newly allocated memory:

The file was in its mapped format (reference), and for some reason i was unable to unmap it to its raw format, trying various methods. i suspect that the reason is because the PE headers were corrupted in some way. So in some point i gave up the unmapping, and moved on to the very JUICY strings armed with my prior knowledge on Conficker actions.

First were the `autorun.inf` strings which were written to every Disk on Key that inserted to an infected machine:

```
useautoplay=1
icon
action
.\%s\%s\%s.%s,%s
%s\%s
%s%s
%s%s\%s\%s.%s
S-%d-%d-%d-%d%d%d-%d%d%d-%d%d%d-%d
RECYCLER
```

Then there are a big list of security producs and related names, which will compared against each DNS lookup the host makes, and if the DNS request contains any of these words, the request will be blocked! that is done by hooking the DNS library in every process!!

clamav

comodo

quickheal

avira

avast

esafe

ahnlab

centralcommand

drweb

grisoft

nod32

f-prot

jotti

kaspersky

f-secure

computerassociates

networkassociates

etrust

panda

sophos

trendmicro

mcafee

norton

symantec

defender

rootkit

malware

spyware

virus

It also has the ability to retrieve the external IP address of the machine by quering each of those sites:

http://www.getmyip.org
http://www.whatsmyipaddress.com
http://www.whatismyip.org
http://checkip.dyndns.org
http://%d.%d.%d.%d:%d/%s

And there is the password list (part of it, it's longer):

```
super
secret
backup
manager
ihavenopass
nothing
nopassword
nopass
love123
home123
qwe123
pw123
root123
pass123
pass12
pass1
admin123
admin12
admin1
password123
password12
password1
```

Spreading

The worm spreads itself by 3 mechanisms:

1. By Brute Forcing SMB shares using the password list. when it guesses the right password, it writes the payload to the remote share and runs it by creating a remote service.
2. By Infecting DOKs and removable drives.
3. By ms08-067, which is being exploited heavily by it. for that, the worm creates a local HTTP server on the infected machine, which serves the payload for any host that is exploited successfuly.

More capabilities which not discussed

1. The worm contains a DGA algorithm (explained here).
2. The worm changes TCP settings, like the allowed current TCP connections, in order to optimize the Brute Force process.
3. The worm shuts down system services, like `Windows Defender` and `Background Intelligent Transfer Service` to disrupt automatic updates and protections.
4. The worm injects itself to system services like `Explorer.exe` and `Svchost.exe` .
5. The worm deletes the System Restore Points.
6. The worm contains anti-analysis, anti-sandbox and anti-vm capabilities, and a lot of obfuscation and "spaghetti code".

Conclusion

Conficker is a sophisticated, contagious, brutal and noisy Windows worm. In this writeup i discussed only a small part of Conficker whole story, there is a <u>comprehensive article</u> about it as well.

Hope you enjoyed :)