DFIR - Identifying The Infection Vector

By Chan YueMeng

Contents

Introduction

Hi all. My name is Chan YueMeng and I currently work in a MNC as a Cyber Incident Response Analyst.

After coaching from my peers, seniors and good books regarding DFIR, I start to appreciate the value of performing DFIR when a cyber incident happens and I feel good after uncovering artifacts on host that cause the incident or malicious payload unpack from the initial payload which has no signature from our AV at that moment in time.

I start to look into and explore using alternate open source tool beside the expensive paid tool use in my company in which I would like to share in this presentation on uncovering the infection vector of a cyber incident.

Current practice of DFIR usually covers the infection vector, scope, remediation and containment. This writeup is just part of the process which covers mainly on the infection vector in which i believe the information uncover could help to expand the scope, take action on the right remediation as well as contained the incident.

This also helps to answer the following following question as mention below

1)How the malware got in?
2)How the malware remains on the host?
3)How does it spread?

The infection vector part is usually the second thing I will do when an incident was raised after I have verified it on the indicators provided either by network or host logs provided.

Currently the most common infection vector is usually either by web, email and external devices like thumb drive.

This presentation was created which relates to an incident which was originally raised due to a DNS callout to a certain malicious domain observed in the DNS log.

I need to verify what actually happen on the host that cause the callouts and I would like to share on how I make use of open source tools to get the necessary artifacts, analysing the results to uncover the infection vector as well as uncover anti-forensic attempt to prevent me from doing a timeline analysis on the $MFT.

Please note that the following methods indicate in this presentation might not be consider forensic sound and instead it is more focus on collecting and analysing a malware incident event.

So here it goes…

Getting the relevant artifacts using ftkimager lite

When incident happen, I usually want to know the following information such as

Is there persistence?

what is the date and time of the suspicious file created on the file system?

what was created on the file system around that time when the first indication was observed?

what type of file related activities was observe when the suspicious file was created on host?

Below are the usual artifacts I acquire when doing the first triage

$MFT

NTUSER.dat in each profile found (if host is a share laptop or PC)

SYSTEM hive

SOFTWARE hive

Look for persistence using rip.pl

I would check for persistence first in all the user profile NTUSER.dat found on host.

```
rip.pl -r NTUSER.DAT -p user_run
```

And it happens that one of the profile contain a suspicious key value mention in below key

```
Software\Microsoft\Windows\CurrentVersion\Run
```

Key value

```
ALERT: run: temp found in path: c:\users\[redacted]\appdata\local\temp\launch.exe
```

Check out the folder contents that holds the suspicious malicious exe if found

From above findings, I would check the folder c:\users\<user>\appdata\local\temp\ visually if possible.

Follow by that, I would sort the timestamp to verify if there are additional files created visually

Please take note that the timestamp visually seen is $SIA ($Standard_Information Attribute) time

Coincidentally, I found the following file call jucheck.exe having the same SIA timestamp.

Using log2timeline to extract timeline from $MFT

Base on above findings, i use log2timeline to parsed information out from $MFT so that I could verified the timeline of the file jucheck.exe and launch.exe.

```
log2timeline -z UTC -f mft -d -w hitcon_demo \$MFT
```

The above command will parsed both $SIA ($Standard_Information attribute) and $FNA ($File_Name attribute) information on the files, folders etc residing on the host.

After the file hitcon_demo is created, i will try to grep both jucheck.exe and launch.exe

```
grep '\bjucheck.exe\b' hitcon_demo
```

```
11/02/2014,16:12:55,,MA.B,FILE,NTFS $MFT,$SI [MA.B] time,-,-,/Users/[redacted]/AppData/Local/Temp/jucheck.exe,
,Log2t::input::mft,-
02/27/2015,09:22:23,,MACB,FILE,NTFS $MFT,$FN [MACB] time,-,-,/Users/[redacted]/AppData/Local/Temp/jucheck.exe,
,Log2t::input::mft,-
```

```
egrep '\blaunch.exe\b' hitcon_demo
```

```
11/02/2014,16:12:55,,MA.B,FILE,NTFS $MFT,$SI [MA.B] time,-,-,/Users/[redacted]/AppData/Local/Temp/launch.exe,
g2t::input::mft,-
05/12/2015,06:40:27,,MACB,FILE,NTFS $MFT,$FN [MACB] time,-,-,/Users/[redacted]/AppData/Local/Temp/launch.exe,
g2t::input::mft,-
```

There is a reason why I need both the $SIA and $FNA timestamp because there should be no reason why a legitimate file would have timestomp event in the $MFT unless it is attempting to throw us analysis off the target when doing timeline analysis in which this firmly confirm my suspicion that this file is indeed malicious

Later I will make use of windows file system tunneling to demo the difference between $SIA and $FNA timestamp

```
11/02/2014,16:12:55,,MA.B,FILE,NTFS $MFT,$SI [MA.B] time,-,-,/Users/[redacted]/AppData/Local/Temp/jucheck.exe,
,Log2t::input::mft,-
02/27/2015,09:22:23,,MACB,FILE,NTFS $MFT,$FN [MACB] time,-,-,/Users/[redacted]/AppData/Local/Temp/jucheck.exe,
,Log2t::input::mft,-
```

```
11/02/2014,16:12:55,,MA.B,FILE,NTFS $MFT,$SI [MA.B] time,-,-,/Users/[redacted]/AppData/Local/Temp/launch.exe,
g2t::input::mft,-
05/12/2015,06:40:27,,MACB,FILE,NTFS $MFT,$FN [MACB] time,-,-,/Users/[redacted]/AppData/Local/Temp/launch.exe,
g2t::input::mft,-
```

from above result, do you see the differences in the timing?

The $SIA time for jucheck.exe is

11/02/2014 16:12:55 UTC

The $SIA time for launch.exe is

11/02/2014 16:12:55 Now UTC

Both have the same $SIA timestamp.

Now for the $FNA timestamp for jucheck.exe is

02/27/2015 09:22:23 UTC

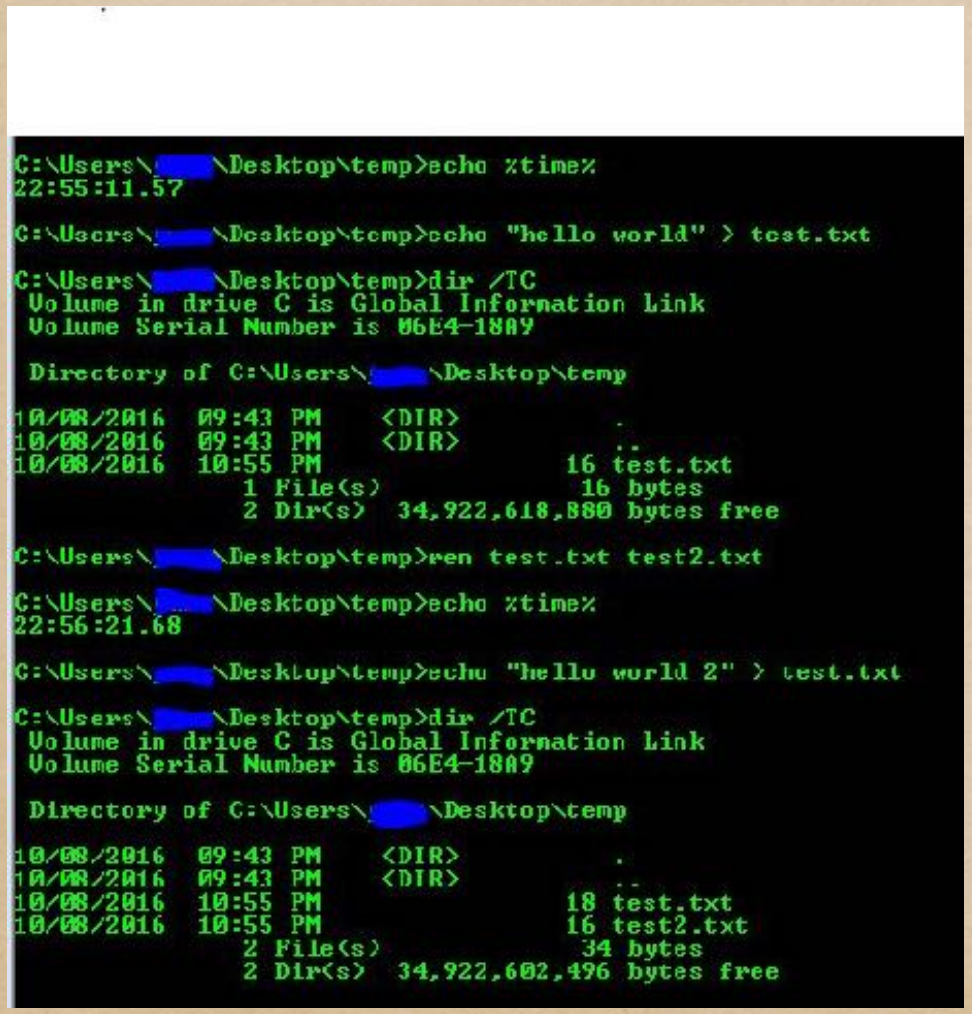
And the $FNA timestamp for launch.exe is

05/12/2015 06:40:27 UTC

From above information, I will use the $FNA timestamp to carry on my timeline analysis because the $SIA timestamp is easily modifiable in which i will explain why later.

What is timestomp

Below is a demo of the difference between $SIA and $FNA using Windows time tunneling feature (Below timing is in SGT)



At about 14:55:11:57 UTC on the 10/08, i create a file call test.txt

Using dir /TC as shown above, the time is indicated as creation time for $SIA timestamp

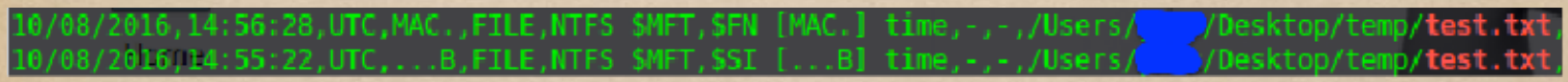
Then I rename the file test.txt to test2.txt at 1 min later (14:56:11:57 UTC)

Within a time frame of 15 sec, i create another file with the same name as the previous file call text.txt

If we run dir /TC, we should see the creation timestamp reflected as 14:56 UTC but instead I am seeing the creation timestamp reflected as 14:55 UTC as shown above.

Please again do note that the timestamp we are seeing in above demo reflects the $SIA timestamp.

Analysing on the $MFT relating to this file tunneling demo as shown below
We can see that $FNA timestamp for test.txt is 14:56:28 UTC 10/08/2016 in which it is the correct time the file is created



Whereas the $SIA timestamp for test.txt is 14:55:22 UTC 10/08/2016.

This method is frequently used by attackers to confuse analyst when doing timeline analysis.

Checking for sign of timestomp

By analysing both the $FNA and $SIA timestamp for both jucheck.exe and launch.exe, it clearly indicate that there is timestomp attempt by the attacker to throw the analyst off the target when doing timeline analysis.

Can you imagine if we use the $SIA time which is 11/02/2014 16:12:55 UTC instead of either $FNA timestamp (02/27/2015 09:22:23 UTC or 05/12/2015 06:40:27 UTC)?

Check for file activity related to web or email on $MFT

Focusing on the $FNA timestamp, I would use the earliest $FNA timestamp found as this earliest time will provide a preferable indication on how the malware got into the host.

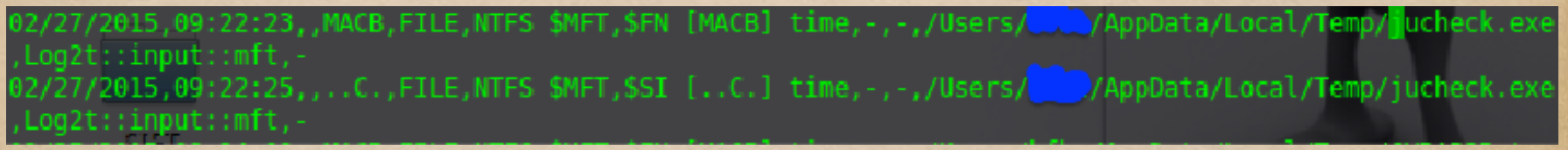
Using the $FNA timestamp at 02/27/2015, I would grep all lines that have this date from the result of the log2timeline

```
grep '02/27/2015' hitcon_demo > timeline
```

Then I will sort according to time.

```
sort -t ',' -k 2,2 timeline > timeline_sorted
```

Usually if the malware is introduced into the host by either web or email, there should be some sign of file activities relating to both web and email that contain for example ‘internet explorer’ or ‘outlook’ (depending on host browser and email client application) prior to the creation of the suspicious exe on the file system.



For this incident, there are none which cause me to focus my attention to usb devices or network file share which would not write file related activities on the $MFT table.

Check on registry using rip.pl for usb information

First thing in my thoughts process is to check if user double click on anything suspicious on the 27th of Feb 2015 at around 09:22:25 UTC

On the NTUSER.dat, i would check on the user assist key

```
rip.pl -r NTUSER.DAT -p userassist | grep -A1 'Feb 27'
```

```
Fri Feb 27 09:22:14 2015 Z
  G:\Afroman.lnk (0)
```

As we can see from above result, there is a link file click on a G:\ drive on 27th Feb 2015 at 09:22:14 UTC

Now we need to check on the common USB registry setting for any sign that a USB device was plugged into the host on 27 Feb 2015 near the time 09:22:14 UTC

```
rip.pl -r SYSTEM -p usb | grep 'Feb 27'
```

```
rip.pl -r SYSTEM -p usbstor | grep 'Feb 27'
```

```
rip.pl -r SYSTEM -p devclass | grep 'Feb 27'
```

Above keys does not produce any result except below mp2 key where there are result found on Feb 27 2015 but the timing indication is 4 minutes later and there is no indication of the usb device model.

```
rip.pl -r NTUSER.DAT -p mp2 | grep 'Feb 27'
```

```
Fri Feb 27 16:38:50 2015 (UTC)
Fri Feb 27 13:06:33 2015 (UTC)
Fri Feb 27 13:01:10 2015 (UTC)
Fri Feb 27 10:02:39 2015 (UTC)
Fri Feb 27 09:26:14 2015 (UTC)
Fri Feb 27 16:36:50 2015 (UTC)
Fri Feb 27 13:06:33 2015 (UTC)
Fri Feb 27 09:26:10 2015 (UTC)
```

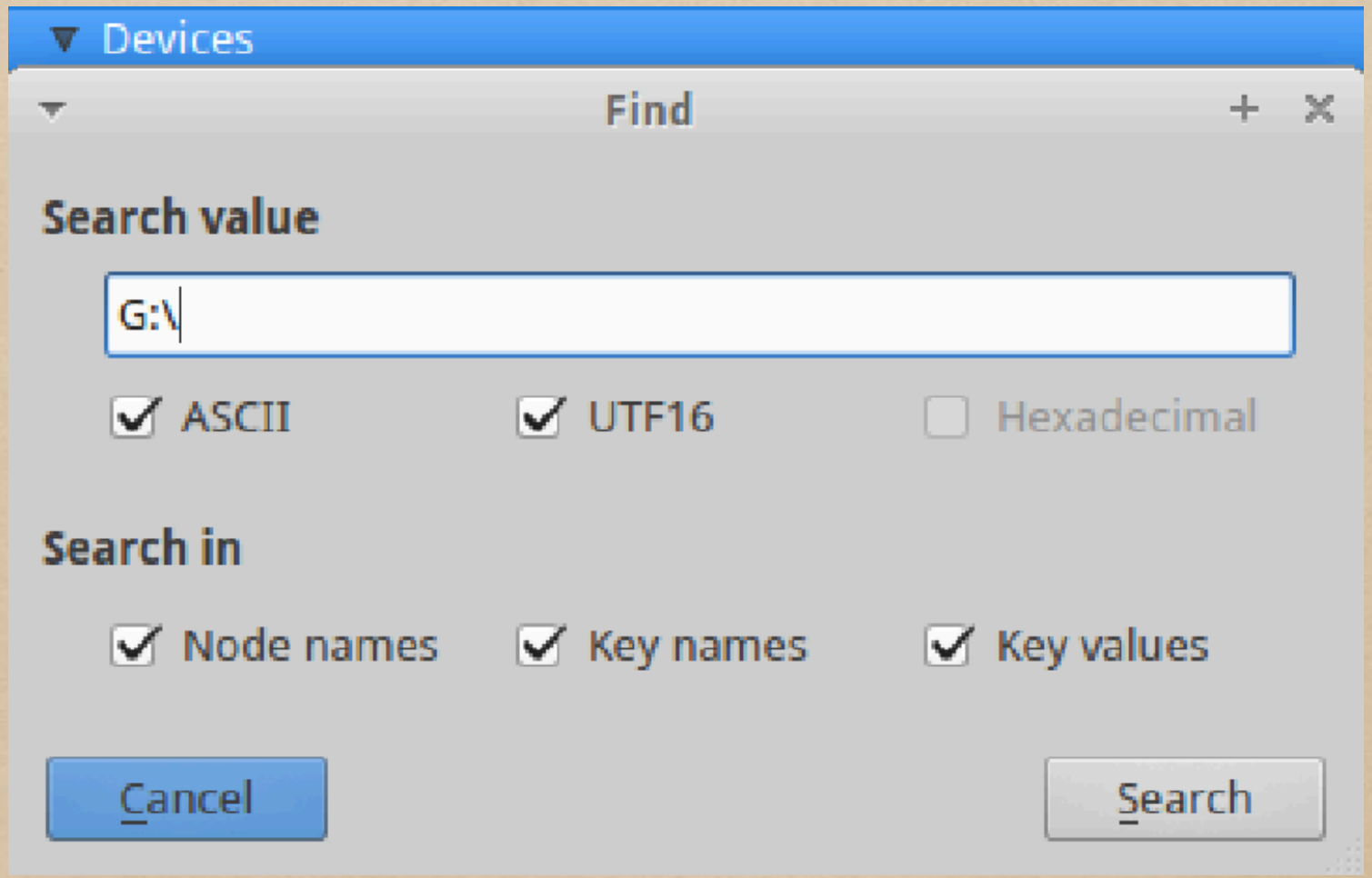
As nothing was found, I resort to using the G:\ drive as a leverage in host SOFTWARE hive

Using fred (Forensic Registry EDitor) on my linux host, I open up the SOFTWARE hive

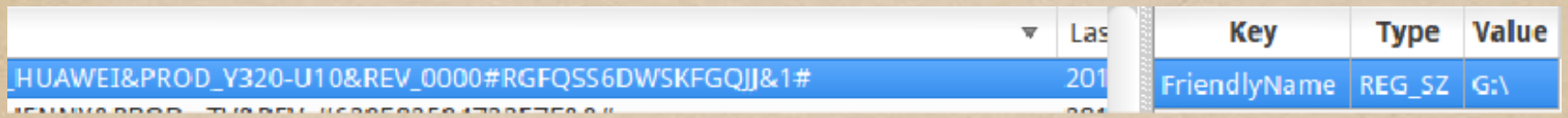
```
fred SOFTWARE
```

```
Forensic Registry EDitor (fred) v0.1.1
```

I access the following key at Microsoft\Windows Portable Devices\Devices and search for G:\ drive as the above key directory keep records of usb devices and its assigned attached drive on host.



And I mange to find one instance of a device being assigned with the G:\



As we can see, it is a huawei device with the serial number ‘RGFQSS6DWSKFGQJJ’ (serial number assigned by vendor usually have ‘&’ as the last second character)

From this findings, we will look in our collected logs for indication that this device was plug into other hosts to cover the scope as well as for sign that G:\Afroman.lnk is residing in the userassist key in other profile ntuser.dat

Depending on company policy, we will advise on the appropriate remediation action base on mobile phone being plugged onto host and advise the user to refrain from plugging their personal mobile onto the host itself.

Last but not least, we will perform a analysis on jucheck.exe to determine what other artifacts are created on host and what it does on host before deciding whether to reimage the host.

References

Windows Forensic Analysis Toolkit, Fourth Edition: Advanced Analysis Techniques for Windows 8 4th Edition - By Harlan Carvey

https://digital-forensics.sans.org/community/downloads

https://code.google.com/archive/p/regripper/downloads

https://www.pinguin.lu/fred

http://windowsir.blogspot.sg/2013/07/howto-determinedetect-use-of-anti.html

https://support.microsoft.com/en-us/kb/172190

https://github.com/log2timeline/plaso/wiki/Using-log2timeline

Questions?

Email: chan.yuemeng@gmail.com

linkedin: https://sg.linkedin.com/in/chanyuemeng

twitter: https://twitter.com/chanyuemeng