Forensics [INFO-Y119]

report

Project :Cyber forensics case study

Authors:
Siéwé Kouéta Anicet : 00364245

Professor: R.M.A
Georgi Nikolov

UNIVERSITÉ LIBRE DE BRUXELLES ULB

December 20, 2020

Contents

List of Figures

1 Introduction

This report explains the way and the steps that allowed us to solve and identify the challenge of the investigation on GAST. This analysis is performed on a sample memory and a hard disk backup. That required a detailed volatility investigation and also some Windows malware analysis skills.

Challenge Questions

1. Give an executive summary of what happened based on your findings.
2. What is the exact timeline of the events that lead to the incident as far as the Windows workstation is concerned?
3. was the workstation's user implicated in the incident, and if so:
a. was he willfully committing sabotage,
b. was he of good faith but negligent, or
c. was he acting responsibly and could not have prevented the incident?
4. what are the elements of evidence you have collected that support your answers to the previous questions?

2 Overview

2.1 Investigation Tools

Throughout this investigation, the main tools used are as follows:

tool name	**version**	**sources**
volatility Framework	2.5	https://www.volatilityfoundation.org/
Autopsy	v4.17.0	https://www.autopsy.com/
Atom	1.51.0	https://atom.io/
IDA64	1.51.0	https://www.hex-rays.com/products/ida/support/download_freeware/
virustotal	- -	https://www.virustotal.com/gui/

2.2 Analysis of the PC Information

The memory analysis was performed on an Ubuntu virtual machine, viewed with VMware on a MacOS laptop.

	Physical PC	**Virtual pc (vmware Fusion 11.02)**
OS	MacOS 10.15.6	Ubuntu 14.04.5 LTS
RAM	16 Go	4 Go
CPU	Intel Core i7 6 cœurs	- -
HDD	1 To	100 Go

2.3 The Target Subject PC Information

The environment and the data about the victim PC are as follows:

OS	Windows 7		
	File Name	**File Type**	**File size**
Memory	memcapture.ad1	Access Data 1	701.3 Mo
	memdump.mem	FoxPro Variable File	1.07 Go
	pagefile.sys	Windows System File	1.46 Go
hard disk	Image_Win_7.E01	Encase Image File Format	16 Go
	Image_Win_7.E02	Encase Image File Format	16 Go
	...	Encase Image File Format	1.57 Go
	...	Encase Image File Format	1.57 Go
	Image_Win_7.E15	Encase Image File Format	1.33 Go

3 The Beginning of the Investigation

It is a question of presenting how the questions were approached and resolved. The questions were not answered in order; they were understood as the survey progressed. This analysis part is written linearly over time to show our approach rather than sorting it by question number.

With the plugin `imageinfo` we determined which operating system was installed on the victim machine. The result in figure 1 reported Windows 7 SP1 x86.

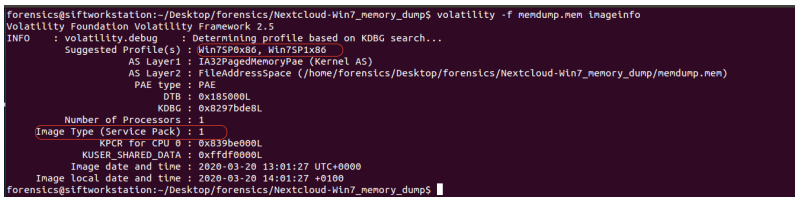


Figure 1: `imageinfo` Result

3.1 Identification of unreliable processes

NOTE: : To better analyze the result of certainly command, the data flow of these have been directly saved in a ***.txt** file.

We first investigated whether there were any known malicious processes. `malfind` allows you to find codes/DLLs hidden or injected into memory.

```
#volatility -f memdump.mem --profile=Win7SP1x86 malfind >| Data/
    malfind.txt
```

The results of figure 2 show us two processes that caught our attention: (**explorer.exe Pid: 5052 lIwRWgTR.exe Pid: 3124**) because although they differ, they have the same signature

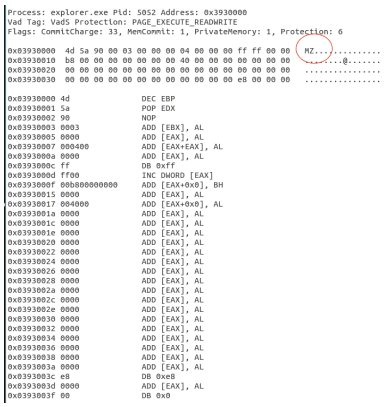


Figure 2: `malfind` Result (explorer.exe Pid:5052)

```
1  Process: lIwRWgTR.exe Pid: 3124 Address: 0x18c0000
2  Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
3  Flags: CommitCharge: 33, MemCommit: 1, PrivateMemory: 1, Protection: 6
4
5  0x018c0000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
6  0x018c0010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00    ........@.......
7  0x018c0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
8  0x018c0030  00 00 00 00 00 00 00 00 00 00 00 00 e8 00 00 00    ................
```

In addition to two other processes identified above, **Process: Dropbox.exe Pid: 3832** also caught our attention.

To verify this, we have, with the plugin `pstree` browsed the list of processes.

```
#volatility -f memdump.mem --profile=Win7SP1x86 pstree >| Data/pstree.txt
```

```
1  Name                                  Pid   PPid   Thds   Hnds Time
2  ------------------------------------------------------------------------------
3  . 0x84b2dd20:lIwRWgTR.exe             3124   3256      4   1161 2020-03-20 09:57:36
4  .. 0x84aea7d8:PING.EXE                2932   3124      0 ------ 2020-03-20 10:23:52
5  .. 0x84be7900:PING.EXE                5564   3124      0 ------ 2020-03-20 10:22:48
6  .. 0x84ae84b8:PING.EXE                3260   3124      0 ------ 2020-03-20 10:23:17
7         /                                       *
8     [snip 93 PING.EXE processes]                *
9         /                                       *
10 .. 0x84b01a10:PING.EXE                1968   3124      0 ------ 2020-03-20 10:24:54
11 .. 0x8431e030:PING.EXE                1628   3124      0 ------ 2020-03-20 10:22:48
12 .. 0x859cfb00:PING.EXE                3628   3124      0 ------ 2020-03-20 10:22:48
13 ... 0x851b86a8:Dropbox.exe            3832   3628     93   1643 2020-03-20 08:06:55
14 .... 0x84beda98:QtWebEnginePro        3212   3832     14    209 2020-03-20 08:08:47
15 .... 0x8433d878:Dropbox.exe           5640   3832      5    227 2020-03-20 08:06:55
16 .... 0x8515fd20:Dropbox.exe           5836   3832      7     79 2020-03-20 08:06:55
```

Note that the suspicious process **lIwRWgTR.exe** (line 3) could be identified between the normal processes and has as child process of **Dropbox.exe** (line 13,15 16).

With `"procdump"` we have dumped the executable of the process **lIwRWgTR.exe Pid: 3124** and **explorer.exe Pid: 5052**.

Figure 3: Caption

```
\$volatility -f memdump.mem --profile=Win7SP1x86 procdump -p 3124,5052 --dump-
    dir=Data/

Volatility Foundation Volatility Framework 2.5
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0x85994030 0x003d0000 explorer.exe         OK: executable.5052.exe
0x84b2dd20 0x00400000 lIwRWgTR.exe         OK: executable.3124.exe
```

Then scan these executables with virustotal 2.1 to get more details about these processes.

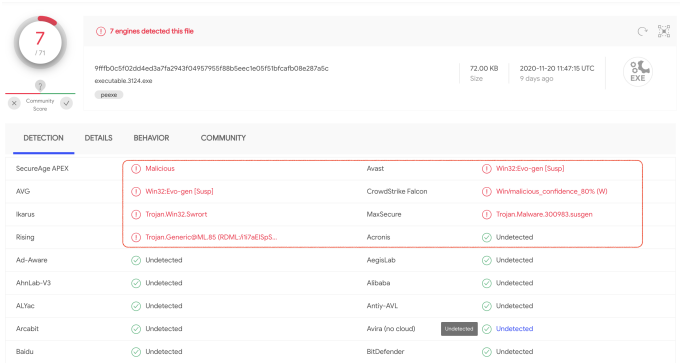


Figure 4: executable.3124.exe (lIwRWgTR) Scan result by virustotal

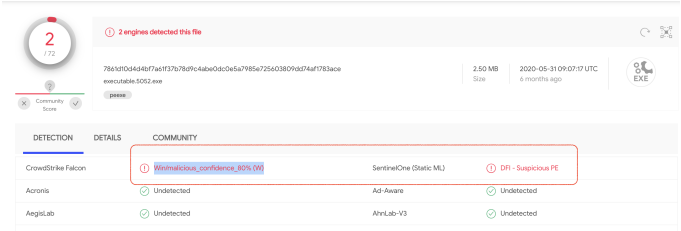


Figure 5: executable.5052.exe (explorer) Scan result by virustotal

The result of figure 4 shows that **lIwRWgTR.exe Pid: 3124** is a malicious program and contains the following malware:

- **Malicious**
- **Win32:Evo-gen [Susp]**:[1](0,), [2](0,) Can Download and install other malware, log keystrokes and sites user visits, send PC information including usernames and browsing history to remote malicious hacker, ...
- **Trojan.Win32.Swrort** [3](0,), [4](0,) may allow a malicious hacker to gain access to the PC
- **Trojan.Generic@ML.85 (RDML:/i1i7aEISpS5vybR** [5](0,) is ransomware used by internet crooks to demand ransom payment from target.
- **malware_confidence_80% (W)** [6](0,) can modify system files, add new folders, create Windows tasks and display advertisements on computers and web browsers.

- **Trojan.Malware.300983.susgen** [7](0,), [8](0,) is a Trojan horse commonly used for spying and distributing ransomware

And those in figure 5 shows that **explorer.exe Pid: 5052** has been infected by two malicious programs:

- **malware_confidence_80% (W)**
- **DFI - Suspicious PE** [9](0,) can also modify system files, create new virus folders and install new Windows services in order to infect and compromise the PC

How hackers got to steal documents, with the plugin `"handles"` we tried to check files handled by malicious processes.

```
$volatility -f memdump.mem --profile=Win7SP1x86 handles -p 3124 -t
    file >| Data/handles_3124.txt
```

handles_3124.txt

```
Offset(V)     Pid      Handle     Access Type            Details
---------- ------ ---------- ---------- ---------------- -------
0x85a3f3c8   3124        0x8   0x120196 File             \Device\NamedPipe
0x8514fd68   3124       0x2c   0x120189 File             \Device\NamedPipe
0x84fad850   3124       0x30   0x120196 File             \Device\NamedPipe
0x84a08bd0   3124       0x3c   0x120189 File             \Device\NamedPipe
0x85c574f0   3124       0x40   0x120196 File             \Device\NamedPipe
0x85badb18   3124       0x68   0x16019f File             \Device\Afd\Endpoint
0x85cef630   3124       0xf4   0x100001 File             \Device\KsecDD
0x84fc7238   3124      0x15c   0x120089 File             \Device\NamedPipe\
0x85bc1908   3124      0x188   0x100080 File             \Device\Nsi
0x8426e6b0   3124      0x190   0x120196 File             \Device\NamedPipe
0x851c5278   3124      0x198   0x120189 File             \Device\NamedPipe
0x858f8998   3124      0x19c   0x120189 File             \Device\NamedPipe
0x841dac68   3124      0x1a0   0x120196 File             \Device\NamedPipe
0x84e47b40   3124      0x1a4   0x120196 File             \Device\NamedPipe
0x851fa388   3124      0x1ac   0x120196 File             \Device\NamedPipe
0x84e25db8   3124      0x1b0   0x120189 File             \Device\NamedPipe
0x8434d3f0   3124      0x1b8   0x120196 File             \Device\NamedPipe
0x850d1280   3124      0x1bc   0x120196 File             \Device\NamedPipe
0x850d19e0   3124      0x1c0   0x120189 File             \Device\NamedPipe
0x85a9b4c8   3124      0x1c8   0x120189 File             \Device\NamedPipe
0x84e25038   3124      0x1d0   0x120189 File             \Device\NamedPipe
0x8586e970   3124      0x1d8   0x120196 File             \Device\NamedPipe
0x85125440   3124      0x1dc   0x120196 File             \Device\NamedPipe
0x859493d0   3124      0x1e0   0x120196 File             \Device\NamedPipe
0x85a5deb8   3124      0x1e8   0x120189 File             \Device\NamedPipe
0x842a33a8   3124      0x1fc   0x120196 File             \Device\NamedPipe
0x8517ad68   3124      0x200   0x120189 File             \Device\NamedPipe
0x84e18038   3124      0x204   0x120189 File             \Device\NamedPipe
0x851d6660   3124      0x208   0x120196 File             \Device\NamedPipe
0x850ff488   3124      0x20c   0x120189 File             \Device\NamedPipe
0x85262bd0   3124      0x210   0x120189 File             \Device\NamedPipe
0x8597bf80   3124      0x214   0x100020 File             \Device\HarddiskVolume1\Users\TimmersVic\Documents\personal
0x85175650   3124      0x228   0x100003 File             \Device\KsecDD
0x85180f80   3124      0x22c   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.7601.24380_none_2b2053e245779417
0x857d3128   3124      0x238   0x120196 File             \Device\NamedPipe
```

Figure 6: `handles` Result for PID=3124

handles_5052.txt

```
Offset(V)     Pid      Handle     Access Type            Details
---------- ------ ---------- ---------- ---------------- -------
0x842f65f0   5052        0x8   0x100020 File             \Device\HarddiskVolume1\Users\TimmersVic\Downloads
0x85990350   5052        0xc   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs
\x86_microsoft.windows.gdiplus_6595b64144ccf1df_1.1.7601.24385_none_5c8ad4dfa00a8e0a
0x85877d90   5052       0x13c   0x100001 File             \Device\KsecDD
0x84fde838   5052       0x144   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.7601.24380_none_2b2053e245779417
0x94389430   5052       0x17c   0x120089 File             \Device\NamedPipe\DropboxDataPipe
0x851ed118   5052       0x1ec   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.7601.24380_none_2b2053e245779417
0x84386778   5052       0x20c   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.7601.24380_none_2b2053e245779417
0x84a2b038   5052       0x224   0x120089 File             \Device\HarddiskVolume1\Windows\System32\en-US\shell32.dll.mui
0x926c0ee8   5052       0x22c   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.7601.24380_none_2b2053e245779417
0x857e0d10   5052       0x24c   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.7601.24380_none_2b2053e245779417
0x84acaaa0   5052       0x28c   0x120089 File             \Device\HarddiskVolume1\Windows\Fonts\StaticCache.dat
0x858c9580   5052       0x294   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.c..-controls.resources_6595b64144ccf1df_6.0.7600.16385_en-
us_581cd2bf5825dde9
0x85abcae8   5052       0x298   0x120089 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.c..-controls.resources_6595b64144ccf1df_6.0.7600.16385_en-
us_581cd2bf5825dde9\comctl32.dll.mui
0x850f9160   5052       0x2b0   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.7601.24380_none_2b2053e245779417
0x858d5a00   5052       0x308   0x100081 File             \Device\HarddiskVolume1\Users\TimmersVic\AppData\Local\Microsoft\Windows\WER\ReportArchive
0x85b870f8   5052       0x30c   0x100081 File             \Device\HarddiskVolume1\ProgramData\Microsoft\Windows\WER\ReportArchive
0x8515d190   5052       0x314   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.7601.24380_none_2b2053e245779417
0x85a3edd0   5052       0x32c   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.7601.24380_none_2b2053e245779417
0x85171880   5052       0x360   0x100081 File             \Device\HarddiskVolume1\Users\Public\Desktop
0x850fd038   5052       0x3cc   0x100081 File             \Device\HarddiskVolume1\Users\TimmersVic\AppData\Local\Microsoft\Windows\Burn
0x8584fd80   5052       0x3d0   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.7601.24380_none_2b2053e245779417
0x85135b30   5052       0x478   0x120189 File             \Device\NamedPipe
0x84347488   5052       0x488   0x120089 File             \Device\HarddiskVolume1\Windows\System32\en-US\UIAutomationCore.dll.mui
0x85b90570   5052       0x488   0x100081 File             \Device\HarddiskVolume1\Users\TimmersVic\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\User Pinned
0x85966e10   5052       0x490   0x100081 File             \Device\HarddiskVolume1\Users\TimmersVic\Desktop
0x859cb650   5052       0x4b0   0x100081 File             \Device\HarddiskVolume1\Users\TimmersVic\AppData\Roaming\Microsoft\Windows\Libraries
0x85bae550   5052       0x4b8   0x100081 File             \Device\HarddiskVolume1\Users\TimmersVic\AppData\Roaming\Microsoft\Windows\Network Shortcuts
0x85ca83b0   5052       0x4c0   0x100081 File             \Device\HarddiskVolume1\Users\TimmersVic\Links
0x859adb40   5052       0x4c8   0x100081 File             \Device\HarddiskVolume1\Users\TimmersVic\AppData\Local\Microsoft\Windows\Burn
0x842cbc08   5052       0x4d8   0x100081 File             \Device\HarddiskVolume1\Users\TimmersVic\Documents\code\project AMAR
0x850968d8   5052       0x4e0   0x100081 File             \Device\HarddiskVolume1\Users\TimmersVic\Documents\code
0x85c6a268   5052       0x52c   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.7601.24380_none_2b2053e245779417
0x851b7338   5052       0x55c   0x12019f File             \Device\WMIDataDevice
0x8598dc78   5052       0x56c   0x100020 File             \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.7601.24380_none_2b2053e245779417
```

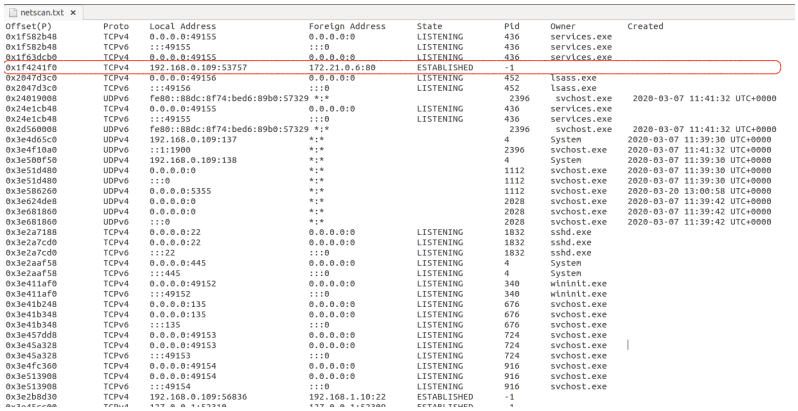
Figure 7: `handles` Result for PID=5052

On the list of files that the process **lIwRWgTR.exe Pid: 3124** was able to handle (figure 6), we notice that the latter had access to the user's personal document *TimmersVic*

Likewise, the figure 7 shows that the process **explorer.exe Pid: 5052** has had access to the personal document of the user *TimmersVic*, particularly the "Project AMAR" folder.

3.2 Network connection analysis

The `"netscan"` plugin helped us to list open network connections on the system.



netscan.txt

Offset(P)	Proto	Local Address	Foreign Address	State	Pid	Owner	Created
0x1f582b48	TCPv4	0.0.0.0:49155	0.0.0.0:0	LISTENING	436	services.exe	
0x1f582b48	TCPv6	:::49155	:::0	LISTENING	436	services.exe	
0x1f63dcb0	TCPv4	0.0.0.0:49155	0.0.0.0:0	LISTENING	436	services.exe	
0x1f4241f0	TCPv4	192.168.0.109:53757	172.21.0.6:80	ESTABLISHED	-1		
0x2047d3c0	TCPv4	0.0.0.0:49156	0.0.0.0:0	LISTENING	452	lsass.exe	
0x2047d3c0	TCPv6	:::49156	:::0	LISTENING	452	lsass.exe	
0x24019008	UDPv6	fe80::88dc:8f74:bed6:89b0:57329 *:*			2396	svchost.exe	2020-03-07 11:41:32 UTC+0000
0x24e1cb48	TCPv4	0.0.0.0:49155	0.0.0.0:0	LISTENING	436	services.exe	
0x24e1cb48	TCPv6	:::49155	:::0	LISTENING	436	services.exe	
0x2d560008	UDPv6	fe80::88dc:8f74:bed6:89b0:57329 *:*			2396	svchost.exe	2020-03-07 11:41:32 UTC+0000
0x3e4d65c0	UDPv4	192.168.0.109:137	*:*		4	System	2020-03-07 11:39:30 UTC+0000
0x3e4f10a0	UDPv6	::1:1900	*:*		2396	svchost.exe	2020-03-07 11:41:32 UTC+0000
0x3e500f50	UDPv4	192.168.0.109:138	*:*		4	System	2020-03-07 11:39:30 UTC+0000
0x3e51d480	UDPv4	0.0.0.0:0	*:*		1112	svchost.exe	2020-03-07 11:39:30 UTC+0000
0x3e51d480	UDPv6	:::0	*:*		1112	svchost.exe	2020-03-07 11:39:30 UTC+0000
0x3e586260	UDPv4	0.0.0.0:5355	*:*		1112	svchost.exe	2020-03-20 13:00:58 UTC+0000
0x3e624de8	UDPv4	0.0.0.0:0	*:*		2028	svchost.exe	2020-03-07 11:39:42 UTC+0000
0x3e681860	UDPv4	0.0.0.0:0	*:*		2028	svchost.exe	2020-03-07 11:39:42 UTC+0000
0x3e681860	UDPv6	:::0	*:*		2028	svchost.exe	2020-03-07 11:39:42 UTC+0000
0x3e2a7188	TCPv4	0.0.0.0:22	0.0.0.0:0	LISTENING	1832	sshd.exe	
0x3e2a7cd0	TCPv4	0.0.0.0:22	0.0.0.0:0	LISTENING	1832	sshd.exe	
0x3e2a7cd0	TCPv6	:::22	:::0	LISTENING	1832	sshd.exe	
0x3e2aaf58	TCPv4	0.0.0.0:445	0.0.0.0:0	LISTENING	4	System	
0x3e2aaf58	TCPv6	:::445	:::0	LISTENING	4	System	
0x3e411af0	TCPv4	0.0.0.0:49152	0.0.0.0:0	LISTENING	340	wininit.exe	
0x3e411af0	TCPv6	:::49152	:::0	LISTENING	340	wininit.exe	
0x3e41b248	TCPv4	0.0.0.0:135	0.0.0.0:0	LISTENING	676	svchost.exe	
0x3e41b348	TCPv4	0.0.0.0:135	0.0.0.0:0	LISTENING	676	svchost.exe	
0x3e41b348	TCPv6	:::135	:::0	LISTENING	676	svchost.exe	
0x3e457dd8	TCPv4	0.0.0.0:49153	0.0.0.0:0	LISTENING	724	svchost.exe	
0x3e45a328	TCPv4	0.0.0.0:49153	0.0.0.0:0	LISTENING	724	svchost.exe	
0x3e45a328	TCPv6	:::49153	:::0	LISTENING	724	svchost.exe	
0x3e4fc360	TCPv4	0.0.0.0:49154	0.0.0.0:0	LISTENING	916	svchost.exe	
0x3e513908	TCPv4	0.0.0.0:49154	0.0.0.0:0	LISTENING	916	svchost.exe	
0x3e513908	TCPv6	:::49154	:::0	LISTENING	916	svchost.exe	
0x3e2b8d30	TCPv4	192.168.0.109:56836	192.168.1.10:22	ESTABLISHED	-1		

Figure 8: `netscan` Result

Figure 8 shows some connections with invalid PIDs and their missing process names. Based on the branch network architecture and the fact that most malware will necessarily have network activity, we have refined our research to only have network connections that are either open, closed, or pending. 'to be closed:

```
1  \$volatility -f memdump.mem --profile=Win7SP1x86 netscan | egrep '(ESTABLISHED
       |CLOSED|CLOSE_WAIT)'
2  Volatility Foundation Volatility Framework 2.5
3  0x1f4241f0  TCPv4   192.168.0.109:53757  172.21.0.6:80          ESTABLISHED  -1
4  0x3e2b8d30  TCPv4   192.168.0.109:56836  192.168.1.10:22        ESTABLISHED  -1
5  0x3e45cc00  TCPv4   127.0.0.1:52310      127.0.0.1:52309        ESTABLISHED  -1
6  0x3e4e3b40  TCPv4   127.0.0.1:52311      127.0.0.1:52312        ESTABLISHED  -1
7  0x3e4ee538  TCPv4   127.0.0.1:52329      127.0.0.1:52328        ESTABLISHED  -1
8  0x3e569520  TCPv4   127.0.0.1:53782      127.0.0.1:53781        ESTABLISHED  -1
9  0x3e670aa8  TCPv4   127.0.0.1:51185      127.0.0.1:51184        ESTABLISHED  -1
10 0x3e74f0f8  TCPv4   192.168.0.109:53719  172.21.0.6:80          ESTABLISHED  -1
11 0x3e78b348  TCPv4   127.0.0.1:57900      127.0.0.1:57901        ESTABLISHED  -1
12 0x3e7c5780  TCPv4   127.0.0.1:52312      127.0.0.1:52311        ESTABLISHED  -1
13 0x3ecf4b78  TCPv4   127.0.0.1:53745      127.0.0.1:53746        ESTABLISHED  -1
14 0x3ed5d480  TCPv4   127.0.0.1:53607      127.0.0.1:53606        ESTABLISHED  -1
15 0x3ee664e0  TCPv4   127.0.0.1:57904      127.0.0.1:57905        ESTABLISHED  -1
16 0x3ef08de8  TCPv4   127.0.0.1:52315      127.0.0.1:52316        ESTABLISHED  -1
17 0x3ef589b0  TCPv4   127.0.0.1:53746      127.0.0.1:53745        ESTABLISHED  -1
18 0x3ef706c0  TCPv4   127.0.0.1:57901      127.0.0.1:57900        ESTABLISHED  -1
19 0x3efa9438  TCPv4   192.168.0.109:57903  192.228.79.201:80      CLOSED       -1
20 0x3f0370c0  TCPv4   127.0.0.1:57905      127.0.0.1:57904        ESTABLISHED  -1
21 0x3f037c10  TCPv4   192.168.0.109:57372  162.125.36.1:443       CLOSE_WAIT   -1
22 0x3f1a8de8  TCPv4   192.168.0.109:53674  108.177.126.109:993    ESTABLISHED  -1
23 0x3f428390  TCPv4   192.168.0.109:52877  52.41.39.5:443         ESTABLISHED  -1
24 0x3f4c1de8  TCPv4   127.0.0.1:52328      127.0.0.1:52329        ESTABLISHED  -1
25 0x3f4f9868  TCPv4   192.168.0.109:53511  162.125.65.3:443       CLOSE_WAIT   -1
26 0x3f5249b8  TCPv4   127.0.0.1:52316      127.0.0.1:52315        ESTABLISHED  -1
27 0x3f52ebb0  TCPv4   192.168.0.109:53592  162.125.65.3:443       CLOSE_WAIT   -1
28 0x3fcc0af0  TCPv4   127.0.0.1:53081      127.0.0.1:53080        ESTABLISHED  -1
29 0x3fd529f8  TCPv4   127.0.0.1:51184      127.0.0.1:51185        ESTABLISHED  -1
30 0x3fd55330  TCPv4   127.0.0.1:57906      239.255.255.250:443    CLOSED       -1
```

```
0x3fd86de8  TCPv4  127.0.0.1:57910        239.255.255.250:443  CLOSED       -1
0x3fd91008  TCPv4  127.0.0.1:52309        127.0.0.1:52310      ESTABLISHED  -1
0x3fda6370  TCPv4  127.0.0.1:53606        127.0.0.1:53607      ESTABLISHED  -1
0x3fdc4ad8  TCPv4  192.168.0.109:57267    34.225.12.148:443    CLOSE_WAIT   -1
0x3fdca4f0  TCPv4  127.0.0.1:53080        127.0.0.1:53081      ESTABLISHED  -1
0x3fdd21f8  TCPv4  127.0.0.1:53781        127.0.0.1:53782      ESTABLISHED  -1
```

From this result, we deduce that the Windows 7 victim machine had the address **192.168.0.109**, because this address is in the range of the intranet subnet of the subsidiary.

We can also see (line N 4) that the central server where the documents and codes are downloaded is **192.168.1.10**, because we have an SSH connection (port 22) on it.

After excluding his two addresses, we had to identify the following addresses:
- `172.21.0.6:80` (line 3)
- `192.228.79.201:80` (line 19)
- `162.125.36.1:443` (line 21)
- `108.177.126.109:993` (line 22)
- `52.41.39.5:443` (line 23)
- `162.125.65.3:443` (line 25)
- `239.255.255.250:443` (line 30)
- `34.225.12.148:443` (line 34)

By analyzing the logs of network equipment, we noticed that the address `172.21.0.6` established several connections with the machine in the branch's subnet (192.168.0.110) (See figure 9)

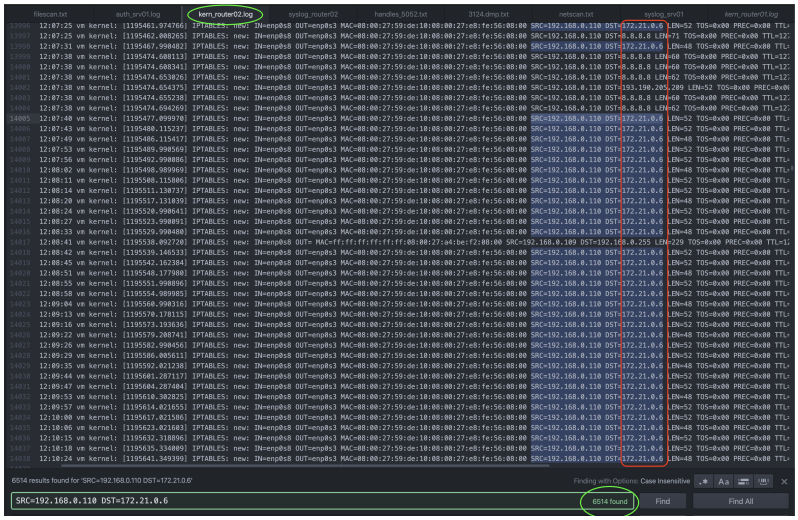


Figure 9: `IP address of C2`

Likewise the address `52.41.39.5` establishes a single connection with the victim machine Windows 7 (192.168.0.109) (See figure 10)

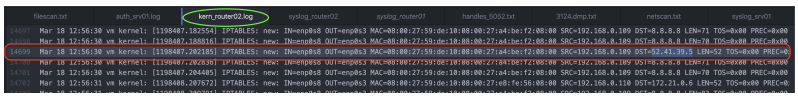


Figure 10: `IP address of second C2`

`192.228.79.201` (line 19) is a DNS server of B.Root-Server-OPS [10](0,), [11](0,).

```
\$ whois 192.228.79.201

refer:          whois.arin.net
inetnum:        192.0.0.0 - 192.255.255.255
organisation:   Administered by ARIN
status:         LEGACY

[snip]

OrgName:          B.Root-Server-OPS
OrgId:            BROOT

[snip]
```

`162.125.36.1` (ligne 21) and `162.125.65.3` (line 25) belong to *Dropbox*

```
\$ whois 162.125.36.1
[snip]

NetRange:         162.125.0.0 - 162.125.255.255
CIDR:             162.125.0.0/16
NetName:          DROPB
NetHandle:        NET-162-125-0-0-1
Parent:           NET162 (NET-162-0-0-0-0)
NetType:          Direct Assignment
OriginAS:
Organization:     Dropbox, Inc. (DROPB)
RegDate:          2015-11-20
Updated:          2015-11-20
Ref:              https://rdap.arin.net/registry/ip/162.125.0.0

[snip]
```

`108.177.126.109` (line 22) Belong to Google, as it connects to port `993` with IMAP protocol [12](0,), we can suggest that it is the Gmail mail server or all other Google service linked to messaging.

```
\$ whois 108.177.126.109

[snip]

NetRange:         108.177.0.0 - 108.177.127.255
CIDR:             108.177.0.0/17
NetName:          GOOGLE
NetHandle:        NET-108-177-0-0-1
Parent:           NET108 (NET-108-0-0-0-0)
NetType:          Direct Allocation
OriginAS:         AS15169
Organization:     Google LLC (GOGL)
RegDate:          2012-03-07
Updated:          2012-03-07
Ref:              https://rdap.arin.net/registry/ip/108.177.0.0
OrgName:          Google LLC

[snip]
```

`52.41.39.5:443` (line 23) and `34.225.12.148` (line 34) were servers hosted on the Ama-

zon cloud

```
\$ whois 34.225.12.148
[snip]

Organization:    Amazon Technologies Inc. (AT-88-Z)
RegDate:         2016-09-12
Updated:         2016-09-12
Ref:             https://rdap.arin.net/registry/ip/34.192.0.0

[snip]
```

3.3 Look for evidence of code injection

As `239.255.255.250:443` is a multicast address, its port number textbf 443, we are somewhat confused.

We used the plugin `"yarascan"` to search the address space for the IP address `239.255.255.250`.

```
$volatility -f memdump.mem --profile=Win7SP1x86 yarascan -Y "
    239.255.255.250" >| Data/yarascan_for_239.255.255.250.txt
```

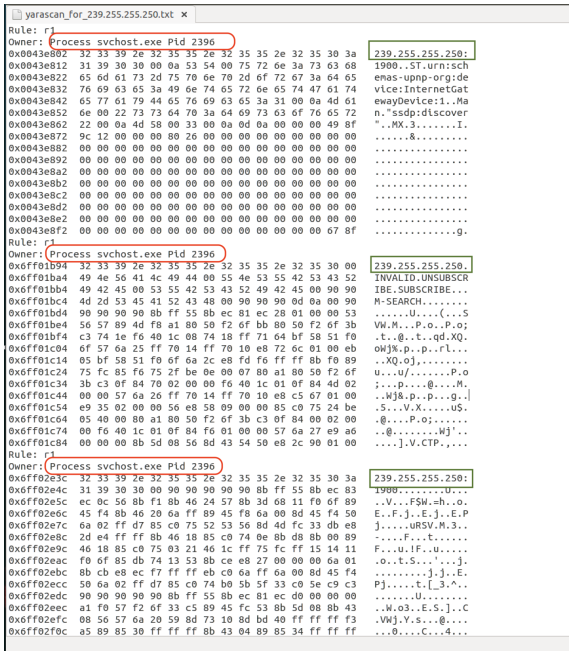


Figure 11: `yarascan` Result for 239.255.255.250

la figure 11 montre que l'adresse IP `239.255.255.250` est présent dans l'espace d'adressage du processuce **`svchost.exe pid: 2396`**

By digging, the process track `svchost.exe pid: 2396` we quickly realize that it leads nowhere. Because in figure 12 there is no parent process **296**. That is suspicious but does not allow us to learn more about the investigation.

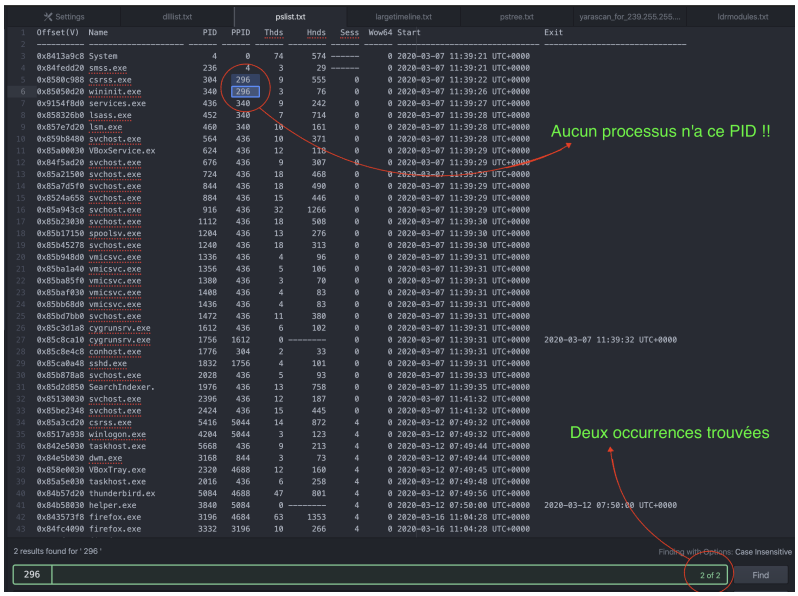


Figure 12: `pid:296 filter on pslist` Result

We therefore continue to explore the process track **lIwRWgTR.exe Pid: 3124** and **explorer.exe Pid: 5052**, by extracting their virtual address space to analyze them.

```
1 \$ volatility -f memdump.mem --profile=Win7SP1x86 memdump -p 3124 - -dump-dir=
      Data/
2 Volatility Foundation Volatility Framework 2.5
3 ************************************************************************
4 Writing lIwRWgTR.exe [  3124] to 3124.dmp
```

From process core dump we have filtered out the keyword "*document*"

```
$strings Data/3124.dmp | egrep 'document'
```

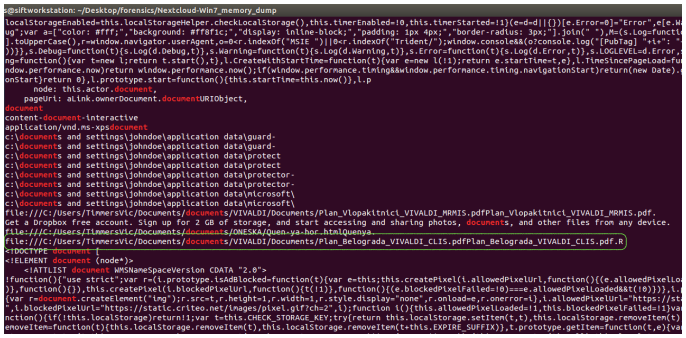


Figure 13: `$strings Data/3124.dmp | egrep 'document'` Result

In the figure 13 the line in green has attracted our attention. We therefore further analyzed this core dump by extracting the strings from the "`3124.dmp`" dump into a file **.txt* [13](0,)

```
$strings -a -n 8 Data/3124.dmp >| Data/3124.dmp.txt
$strings -a -n 8 -el Data/3124.dmp >> Data/3124.dmp.txt
```

In the `3124.dmp.txt` file, we have looked for the IP address `239.255.255.250`. This IP address exists, but with a different port : **`port 1900`** . (figure 14)

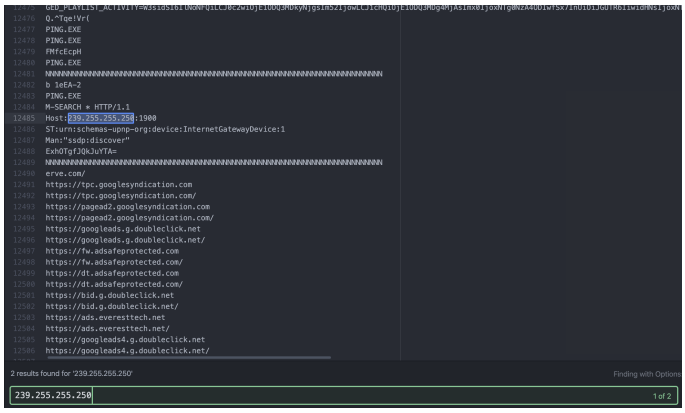


Figure 14: `search 239.255.255.250 in 3124.dmp.tx` Result

Now on Windows systems, the **`1900`** [14](0,) port is associated with the SSDP (Simple Service Discovery Protocol) protocol which allows the discovery of services and information. presence on a network. The address `239.255.255.250` is a multicast address and has been hardcoded in the malicious binary.

Source [14](0,) «... Many devices, including some residential routers, have a vulnerability in the UPnP software that allows an attacker to get replies from port number 1900 to a destination address of their choice ... » We can therefore assume that attackers use this vulnerability. [15](0,),[16](0,)

3.4 Hard disk analysis.

3.4.1 Extraction of objects and data.

as in the paragraph 3.2 (page 10) we had to identify an address as being a Gmail server. We started by extracting emails with Autopsy.

Two email exchanges between **att.ker.1n@gmail.com** (**Attilus**) and **v1c.t1m.m3r@gmai.com** (**Virtoria**) we seem suspicious . Indeed the conversation (figure 22, 23, 24, 25) between the two characters shows that *Virtoria* has downloaded a file ziper **slack .zip**

4 QUESTIONS FROM THE INVESTIGATOR

4.1 summary of the attack

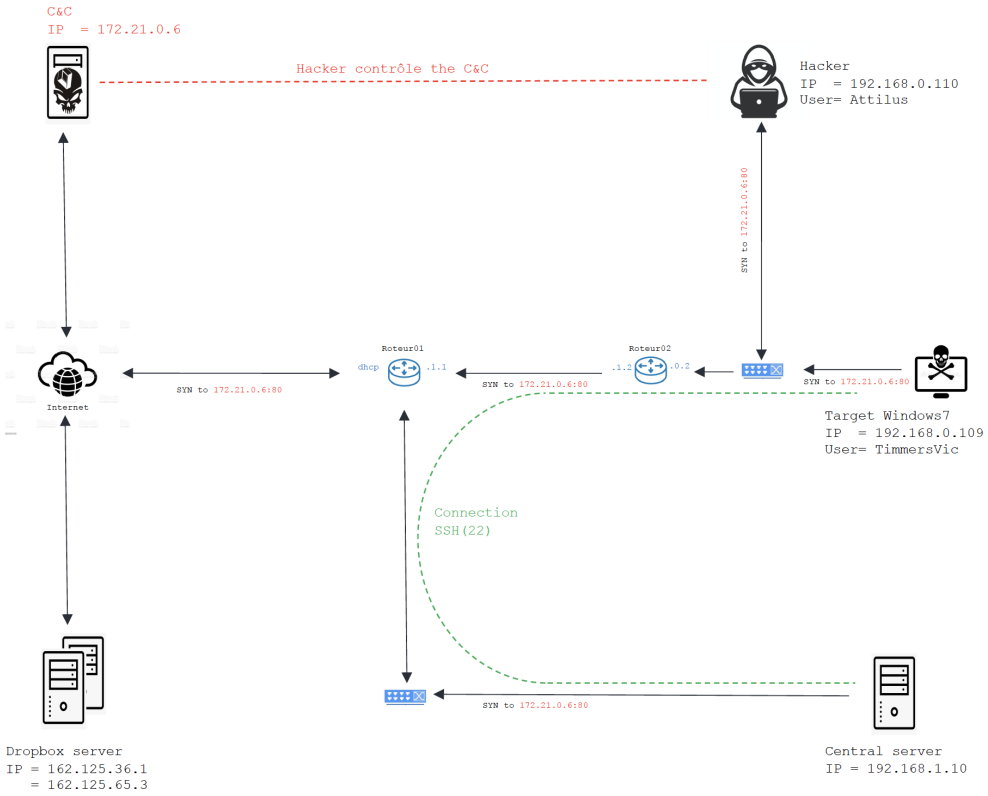


Figure 15: shema of attaker scenario

The Hackers infect the victim's machine via E-mail. They sent malicious software to the victim via WeTranfer. (**slack .zip**). Once the victim has had to unzip the file and open it, the **lIwRWgTR.exe** virus automatically installs itself on the victim's machine. They managed to steal the documents from the central server by connecting via FikeZilla with the victim's identifier (User = "TimmerVic", password = "password1") see figure 29. Once connected, the hackers had to steal the documents of the VIVALDI project by downloading them in the user's directory (See figure 32)

4.2 Event calendar

In this section, we have reconstructed a timeline of the events that occurred on the infecting machine window. Dates in red represent events or items that were intentionally deleted.

- |2020-03-10 at 11:55:04 TimmerVic user creation

- |2020-03-10 at 13:58:04 Mail: New Working station

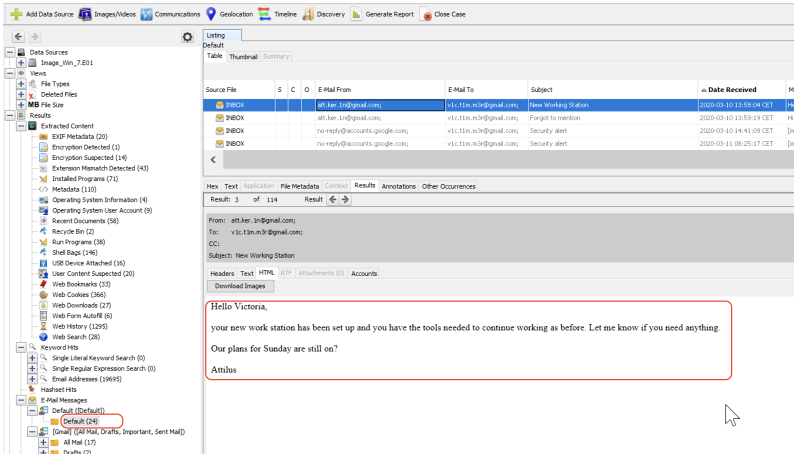


Figure 16: att.ker.1n@gmail.com **to** v1c.t1m.m3r@gmai.com

- 2020-03-10 at 13:59:19 Mail: forgot to mention

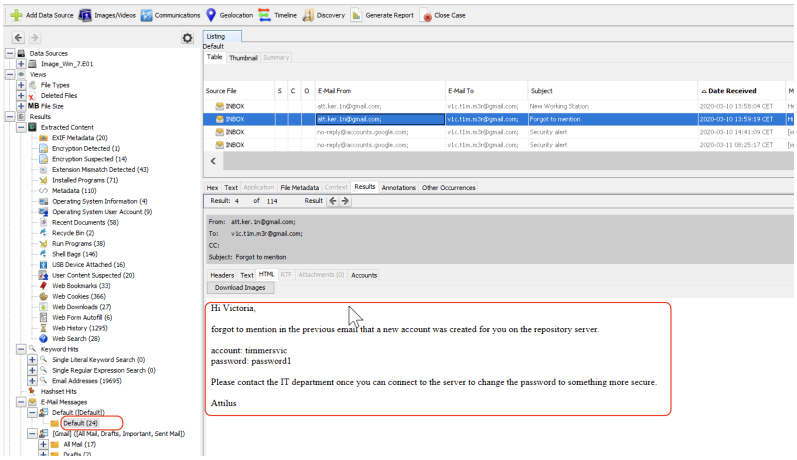


Figure 17: att.ker.1n@gmail.com **to** v1c.t1m.m3r@gmai.com

- 2020-03-11 at 08:31:15 Mail: Re:New Working station

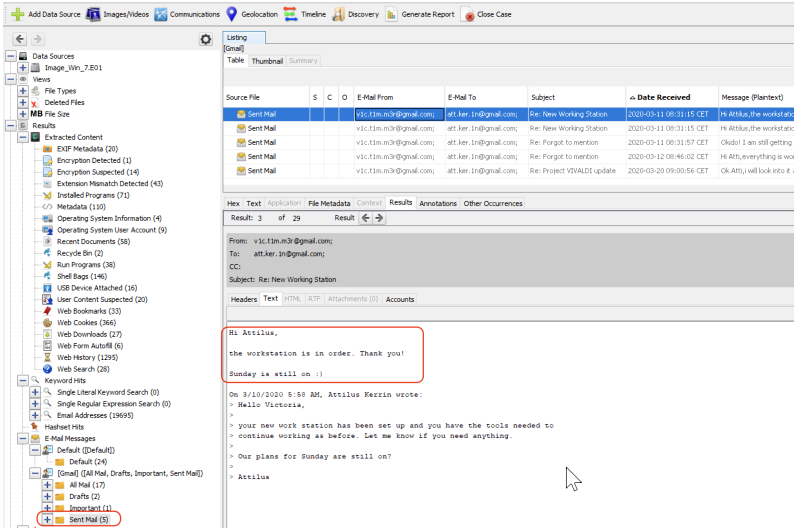


Figure 18: v1c.t1m.m3r@gmai.com **to** att.ker.1n@gmail.com

- 2020-03-11 at 08:31:57 Mail: Re:forgot to mention

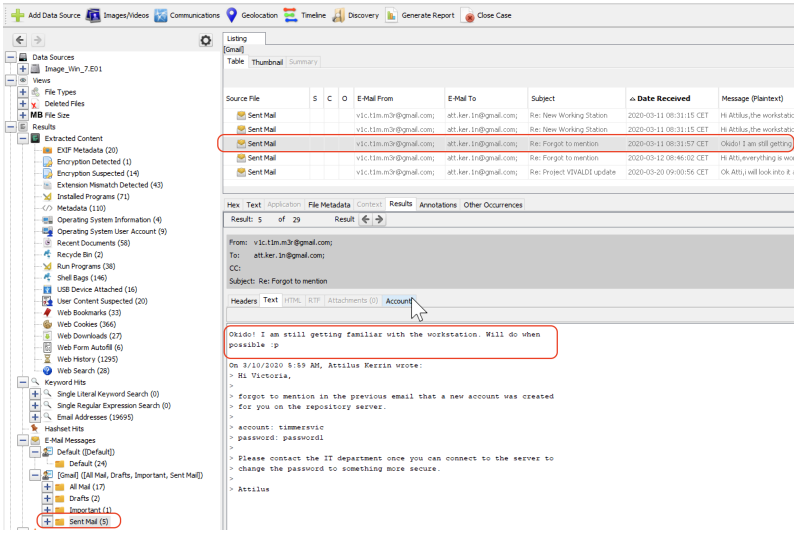


Figure 19: v1c.t1m.m3r@gmai.com **to** att.ker.1n@gmail.com

- 2020-03-12 at 08:46:02 Mail: Re:forgot to mention

Victoria lets Attilus know that she can't run some programs on his machine

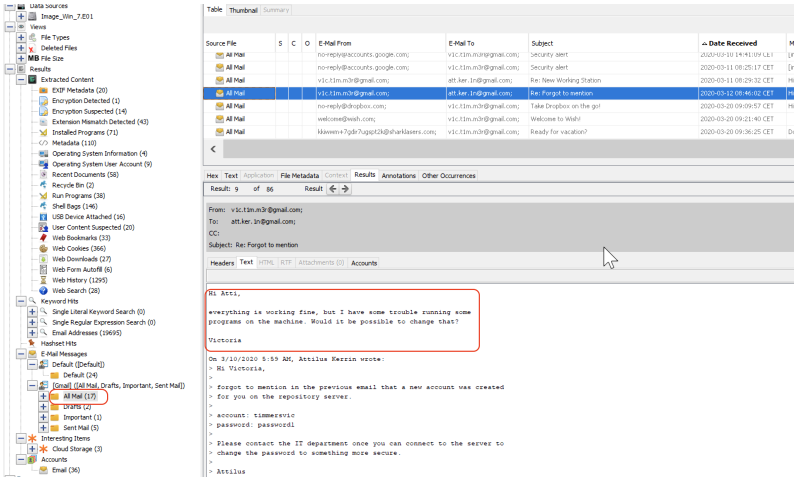


Figure 20: v1c.t1m.m3r@gmai.com **to** att.ker.1n@gmail.com

- 2020-03-12 at 08:46:00 Mail: Re:forgot to mention

Attilus elevated the privileges of the timmervic account to the administrator.

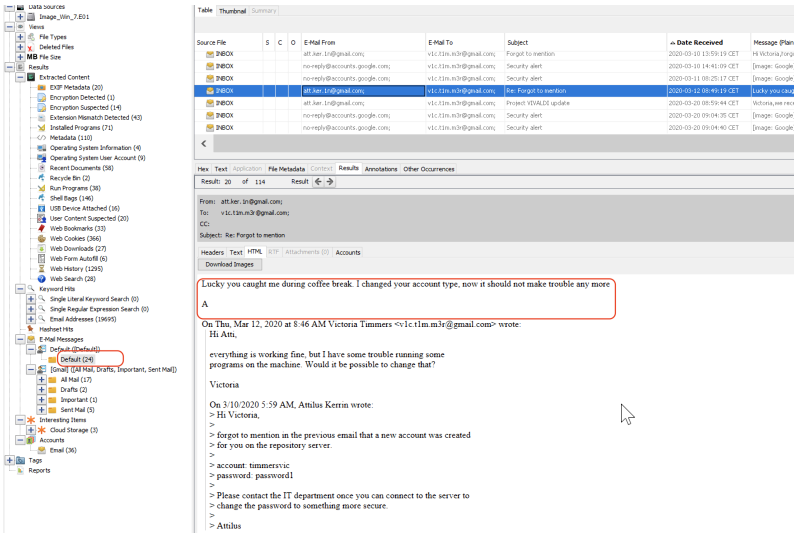


Figure 21: v1c.t1m.m3r@gmai.com **to** att.ker.1n@gmail.com

- 2020-03-17 at 11:48:25 creation of the password file by Victoria

  voire figure 33

- 2020-03-20 at 08:59:44 Mail: project VIVALDI Update

  The attaker wants to exist victoria to download the VIVALDI project with his own username.

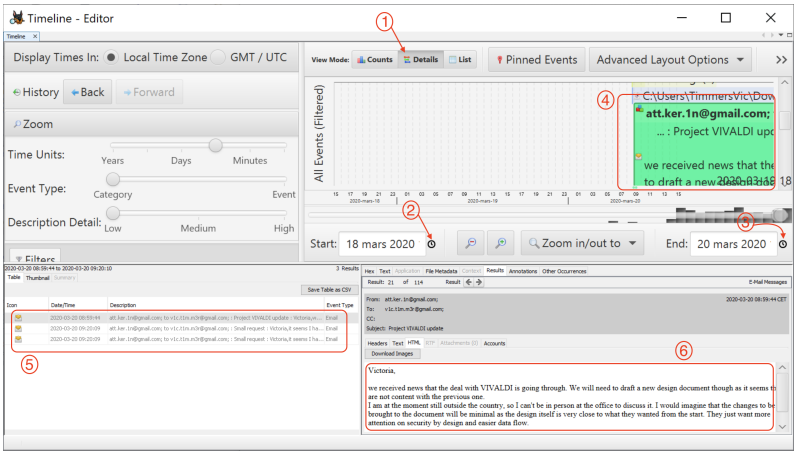


Figure 22: att.ker.1n@gmail.com **to** v1c.t1m.m3r@gmai.com

- 2020-03-20 at 09:00:56 RE:project VIVALDI Update

  Victoria responded to Attilus.

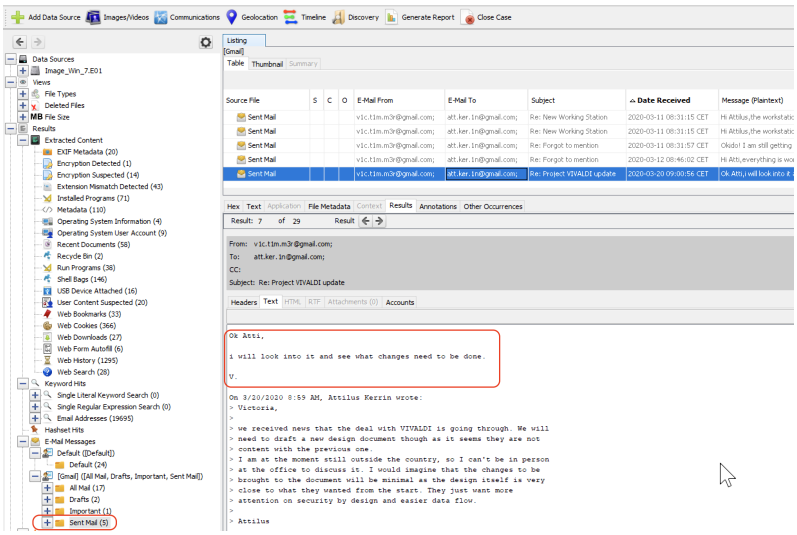


Figure 23: `v1c.t1m.m3r@gmai.com` **to** `att.ker.1n@gmail.com`

- `2020-03-20 at 09:20:03 Mail: Smarl requette`

Attilus fake a request, which will force Victoria to leave her machine unattended for a short time..

- `2020-03-20 at 10:49:51 Mail: useful tool for your`

The attacker makes Victoria believe that he has good tools to help her in her work and offers to send it to her through another channel (WeTransfer).

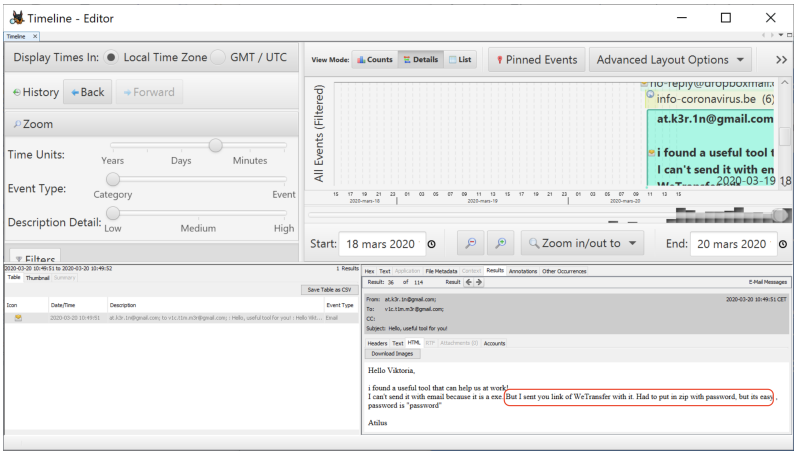


Figure 24: `att.ker.1n@gmail.com` **to** `v1c.t1m.m3r@gmai.com`

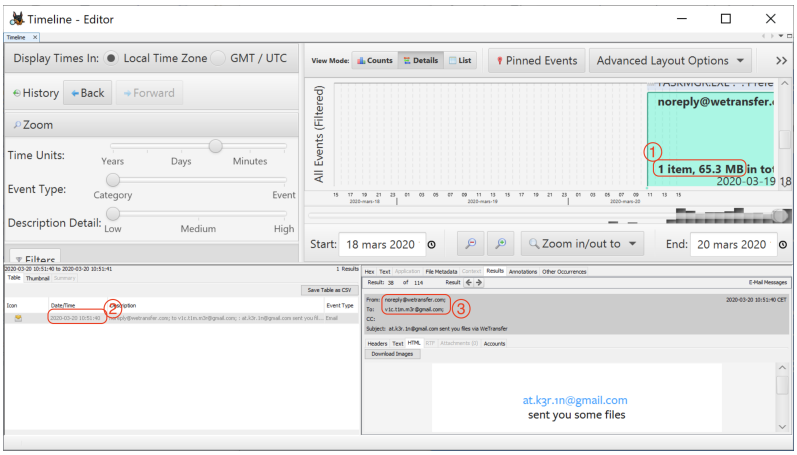


Figure 25: `att.ker.1n@gmail.com(WeTranfert)` **`to`** `v1c.t1m.m3r@gmai.com`

- `2020-03-20 at 10:52:02 downloaded` **`slack.zip`** file

with FTK Image, by exploring the hard disk images provided, we can see that the zip file **slack.zip** contains a malicious executable because the latter has the same **MZ** signature as our **lIwRWgTR.exe Pid: 3124** process. see figure 26

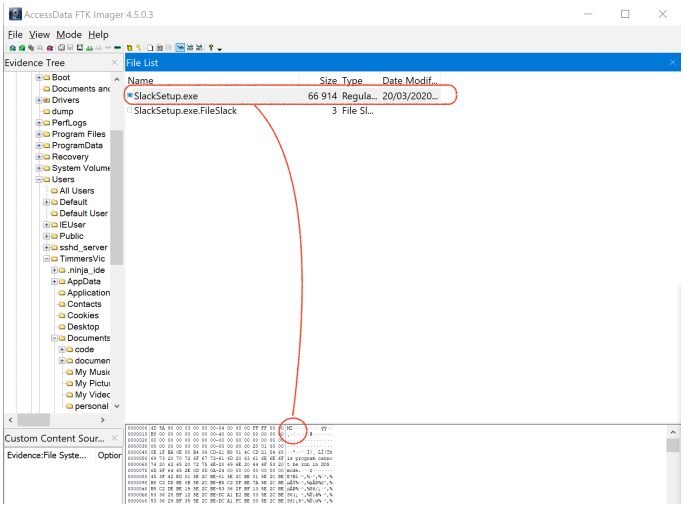


Figure 26: content of the file **slack.zip** download with WeTransfer

- `2020-03-20 at 10:57:35 malware persistence on their target machine`

As most malware tries to hide and achieve persistence on their target machine, in order to run even after restarting the system, we have to check known registry keys to see if the malware does not modify one of the registry keys of one of the Windows AutoStart Extension Points (ASEP).

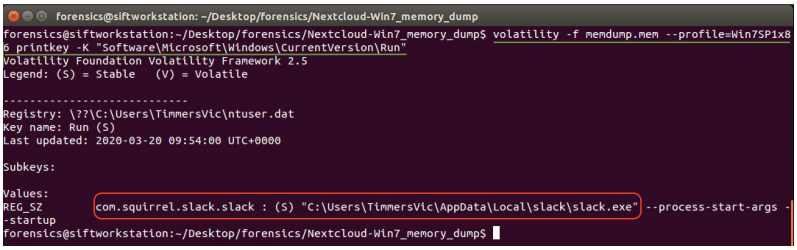


Figure 27: modification ASEP HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run par **slack.exe**

in the figure 27 **slack.exe** is actually in the *RUN*.

and in the `C: \Users\TimmersVic\AppData\Local\Temp\` directory we found the malicious process **lIwRWgTR.exe Pid: 3124**. (see figure 28)

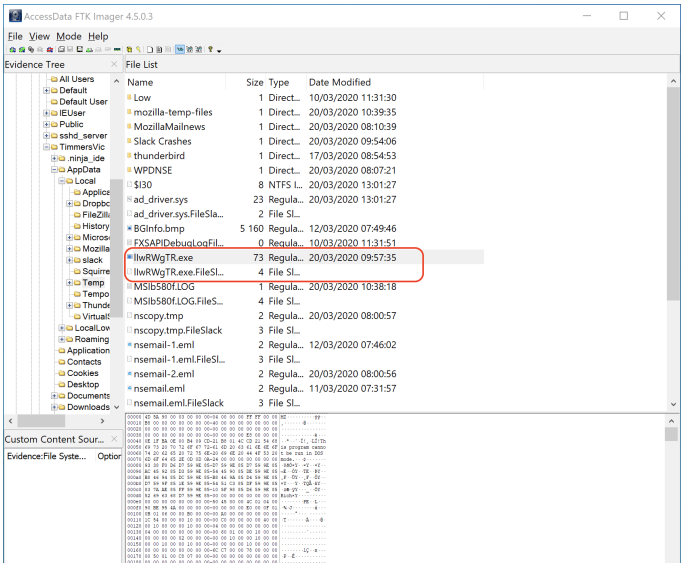


Figure 28: localisation `lIwRWgTR.exe` in the system

- `2020-03-20 at 11:13:26 central server connection`

at 11:13:26 we notice the user **TimmersVic** made a connection to the central server, with the identifiers same identifier which had been entrusted to him

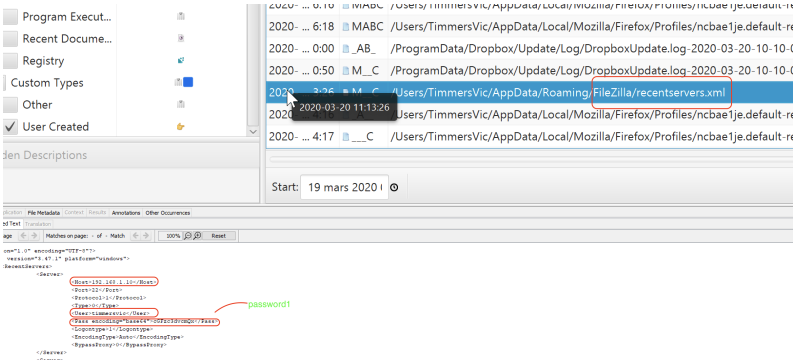


Figure 29: `attempt to connect to the central server with FileZilla`

this connection goes through the router02 at 12:34:39

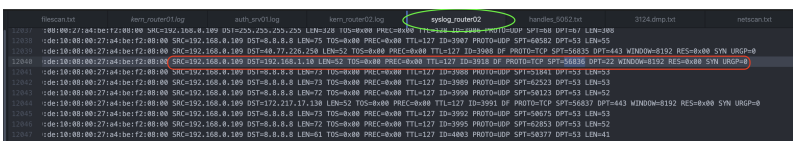


Figure 30: `central server connection`

and is finally accepted at 12:34:40 by the central server

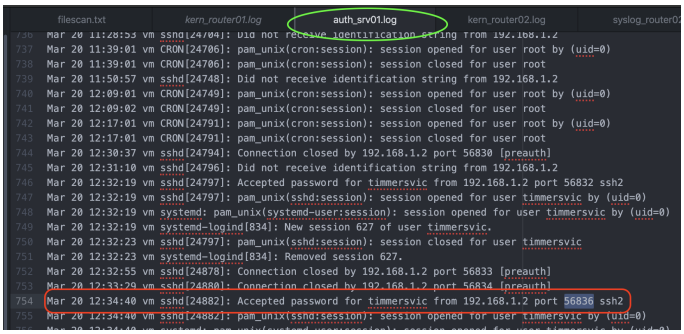


Figure 31: FileZilla connection to the central server

- 2020-03-20 at 13:37:25 central server connection

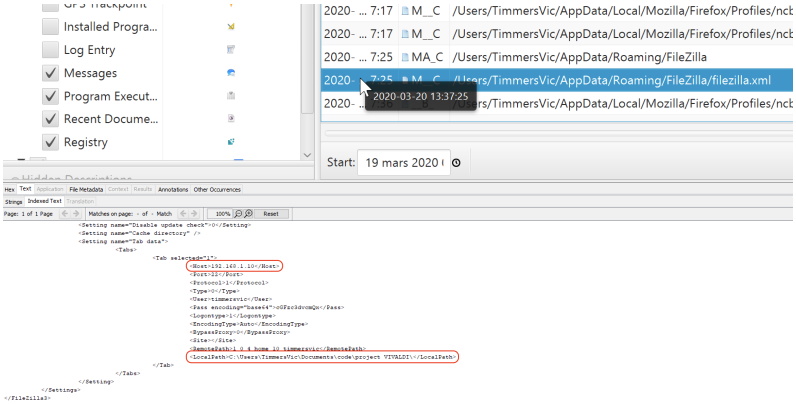


Figure 32: FileZilla connection to the central server

4.3 user involvement

The Windows workstation user was not directly involved. From the conversation in figure 24, it can be deduced that she was negligent and reckless because it is not advisable to install software preventing unknown sources. This recklessness can be found in the fact that she recorded her passwords in *.txt* files (see figure 33)

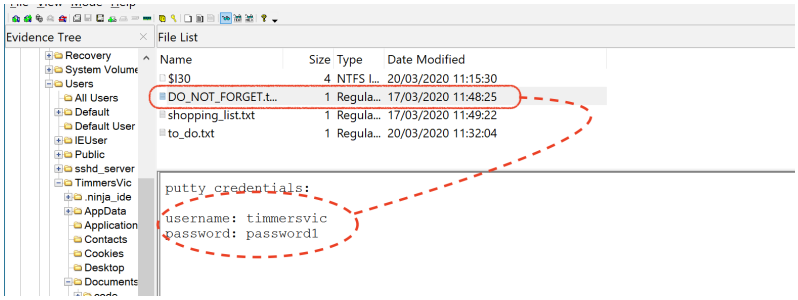


Figure 33: content of DO_NO_FORGET.txt file

In addition, the conversation of figure 18 and 19 shows a more than professional complicity. Because the two plan to meet on a Sunday.

References

[1] Olivia Morelli. (solved) win32:evo-gen explained. thorough removal guide for 2020. https://www.2-spyware.com/remove-win32evo-gen.html, 2020. Accessed on 2020-11-10.

[2] Olivia Morelli. Haal win32:evo-gen weg (verwijdering handleiding) - nov 2020 update. https://zondervirus.nl/win32evo-gen/, 2020. Accessed on 2020-11-10.

[3] Microsoft Defender. Trojan: Win32 / swrort.a. https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Trojan%3AWin32/Swrort.A, 2020. Accessed on 2020-11-10.

[4] Robert Bailey. Trojan: Win32 / swrort.a how to fix guide. https://howtofix.guide/trojanwin32-swrort-a/, 2020. Accessed on 2020-11-10.

[5] Robert Bailey. Trojan.crypt.msil.generic. https://howtofix.guide/trojan-crypt-msil-generic/, 2020. Accessed on 2020-11-10.

[6] Max. How to remove malicious confidence 80% (d) virus. https://www.fixyourbrowser.com/removal-instructions/remove-malicious_confidence_80-d-virus/, 2020. Accessed on 2020-11-10.

[7] Brandon Skies. Trojan.malware.300983.susgen removal. https://howtoremove.guide/trojan-malware-300983-susgen/, 2020. Accessed on 2020-11-10.

[8 joesandbox]. Analysis report securiteinfo.com.trojan.malware.300983.susgen.2470. https://www.joesandbox.com/analysis/277707/0/html, 2020. Accessed on 2020-11-10.

[9]. , 2020. Accessed on 2020-11-10.

[10] University of Southern California. B-root. https://b.root-servers.org/, 2020. Accessed on 2020-11-12.

[11] dnslytics. Ip information 192.228.79.201. https://dnslytics.com/ip/192.228.79.201, 2020. Accessed on 2020-11-12.

[12] siteground. Email protocols - pop3, smtp and imap tutorial. https://www.siteground.com/tutorials/email/protocols-pop3-smtp-imap/, 2020. Accessed on 2020-11-12.

[13] Wiley. The art of memory forensics: Detecting malware and threats in windows, linux, and mac memory. https://www.amazon.com/Art-Memory-Forensics-Detecting-Malware/dp/1118825098. Part II: Windows Memory Forensics, Page 246.

[14] wikipedia. Simple service discovery protocol. https://en.wikipedia.org/wiki/Simple_Service_Discovery_Protocol, 2020. Accessed on 2020-11-12.

[15] Christian Beedgen. Malware faq: Microsoft windows upnp vulnerabilities. https://www.sans.org/security-resources/malwarefaq/win-upnp, 2020. Accessed on 2020-11-12.

[16] speedguide. Port 1900 details. https://www.speedguide.net/port.php?port=1900, 2020. Accessed on 2020-11-12.