MALWARE ANALYSIS

MOHAMMED SHAAD MEHBOOB MATCHESWALA (21065793)

CARDIFF UNIVERSITY

Index.

Analysis of WindowsLiveMessenger.exe using different tools are as below:

1) Using PEiD.exe tool:

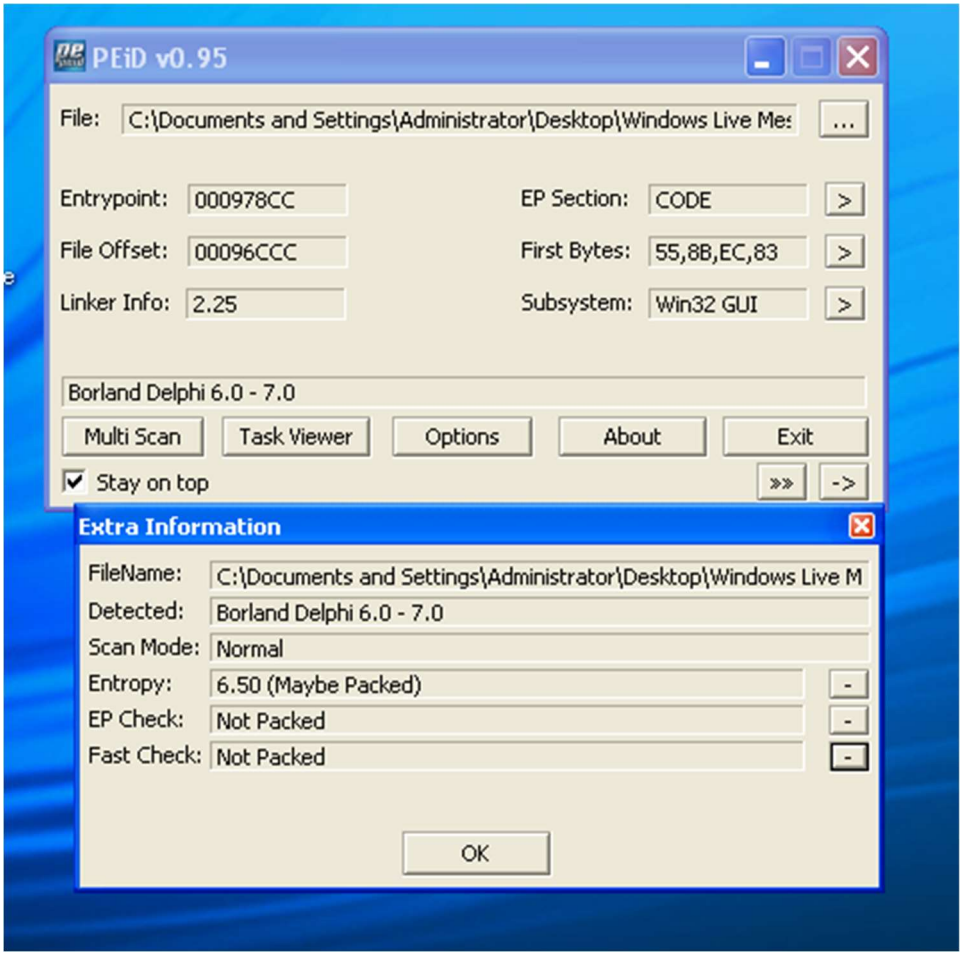


Figure:-1.1

Result: Maybe packed as shown in figure 1.1.

2) Using PEview.exe Tool:

In all the files the Virtual Size of data is smaller or almost the same as the Size of Raw Data as shown in figure 1.2 a, b, c, d, e**.**

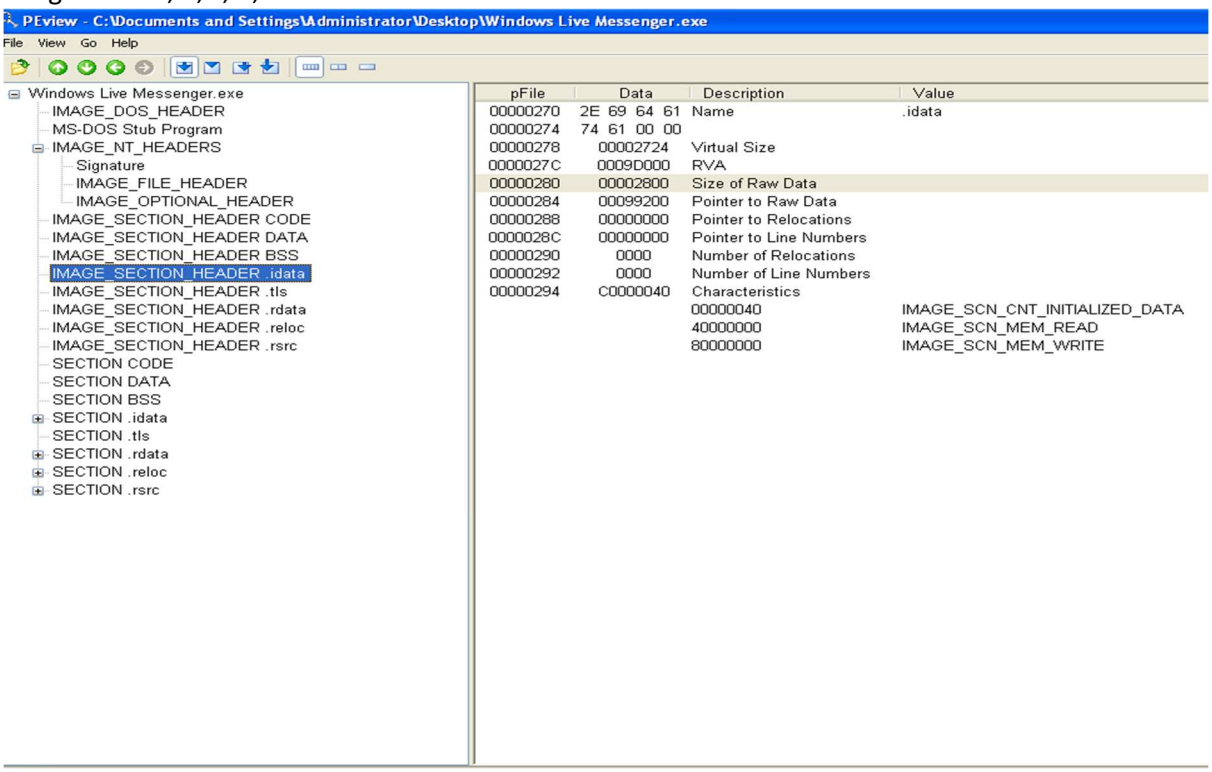


Figure:-1.2-a

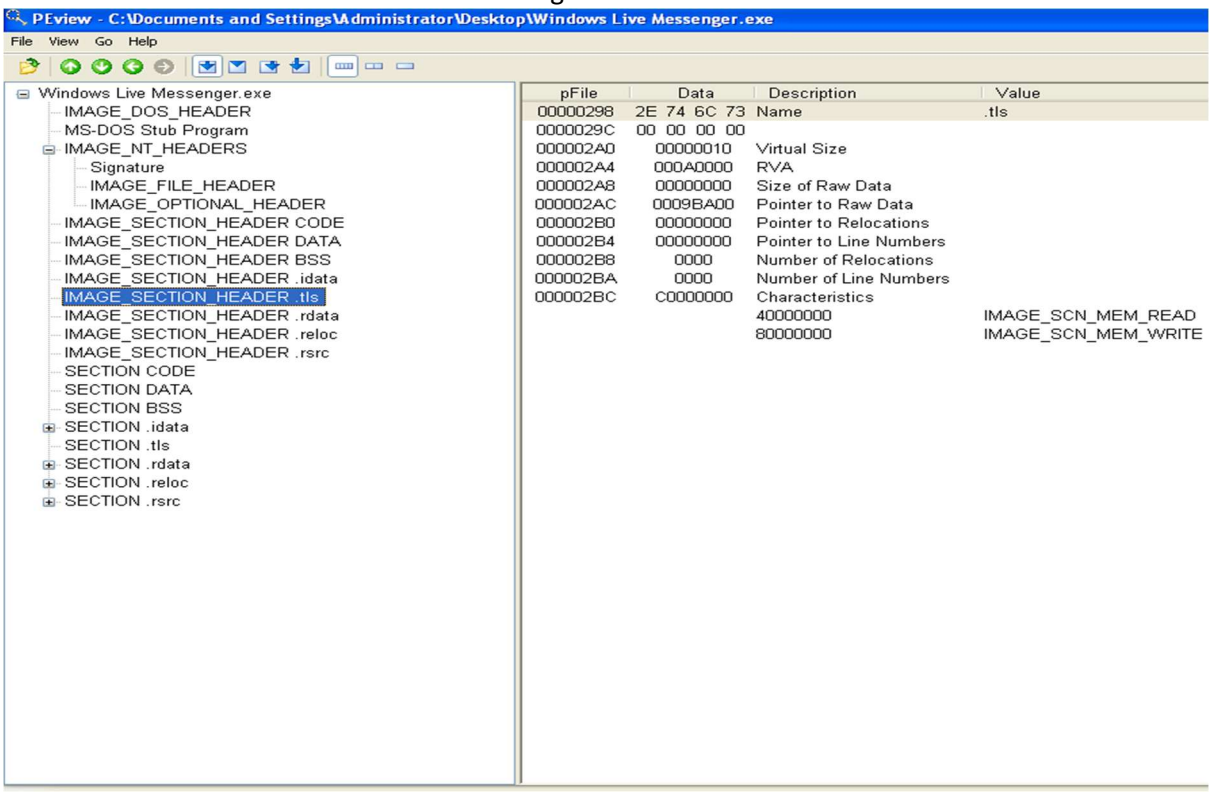


Figure:-1.2-b

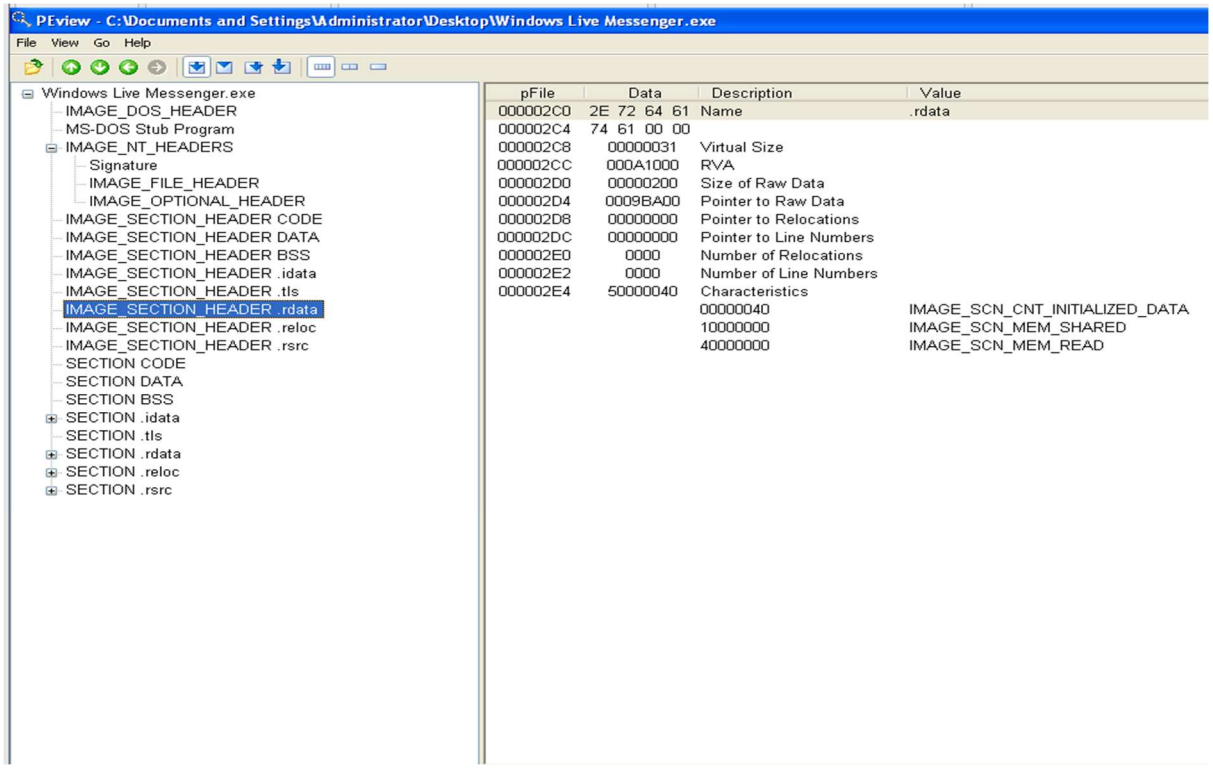


Figure:-1.2-c

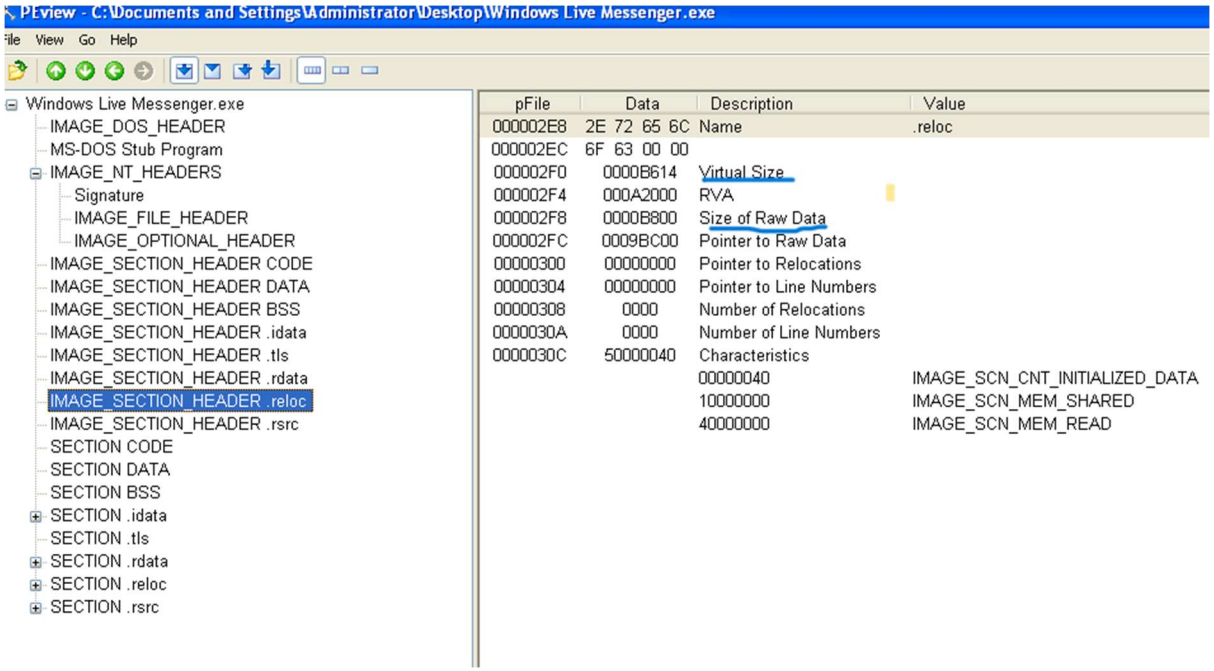


Figure:-1.2-d

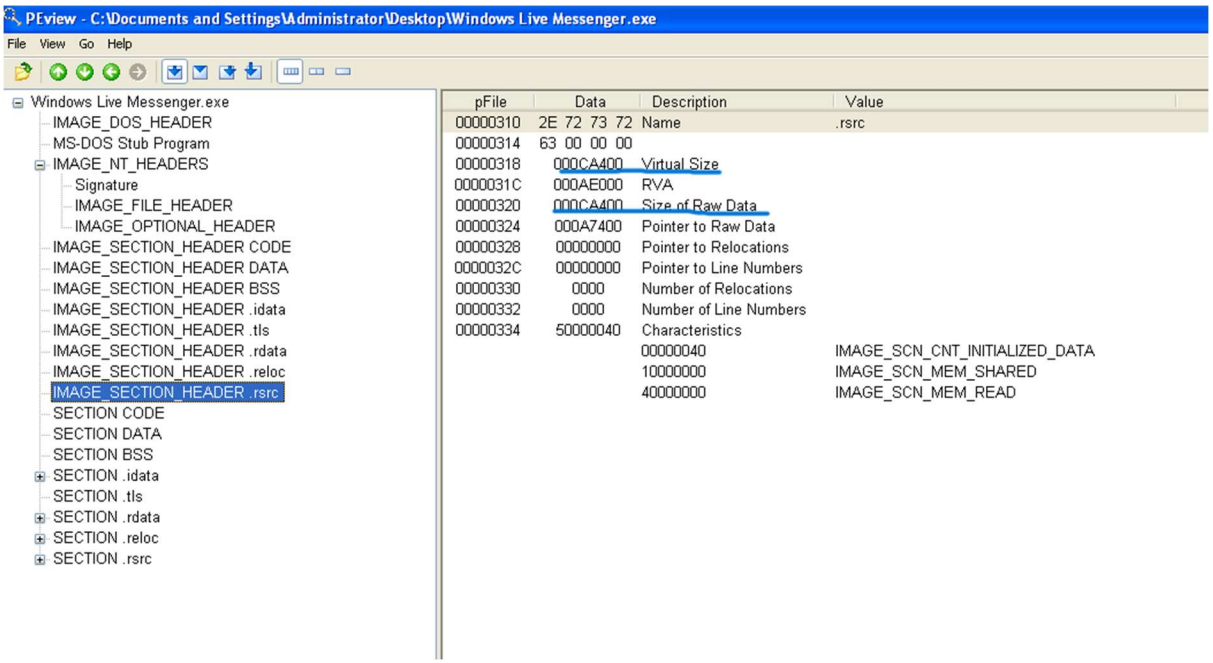


Figure:-1.2-e

Result: The malware is not packed.

Windows live messenger:-

Q1.

The few significant strings are shown in figure 1.3.

REPORT - Notepad

File Edit Format View Help

```
BinText report
------------------------------------------------------------
0000000957C4   0000004963C4      0   yourpassword@password.com/NB
000000095BA0   0000004967A0      0   /pas.txt/HB
000000095BB4   0000004967B4      0   www.ourgodfather.com/NB
0000000960BC   000000496CBC      0   http://status.messenger.msn.com/Status.aspx/NB
000000096114   000000496D14      0   http://get.live.com/getlive/overview/NB
0000000961A8   000000496DA8      0   https://account.live.com/ResetPassword.aspx?mkt=EN-US/NB
000000096824   000000497424      0   C:\Program Files\MSN Messenger\msnmsgr.exe/HB
000000094474   000000495074      0   \Windows Live Messenger.lnk
0000000967BC   0000004973BC      0   msnsettings.dat/HB
00000012F6CE   0000005362CE      0   Bitmap.Data
0000001707D9   0000005773D9      0   Send Password to email:
000000170BF0   0000005777F0      0   Filename: pas.txt
000000170C3A   00000057783A      0   Default Path: C:/
000000170BA5   0000005777A5      0   Save Password in:
000000170AE8   0000005776E8      0   !Create real shortcut on terminate
00000007DC5E   00000047E85E      0   ftpTransfer
000000084C08   000000485808      0   base64
0000000952D1   000000495ED1      0   MailMessage"
00000012F1A7   000000535DA7      0   RequestRemoteAssistants1
00000012F1C9   000000535DC9      0   Request Remote Assistants
000000077520   000000478120      0   LOCALHOST
000000077534   000000478134      0   127.0.0.1
00000007F948   000000480548      0   0.0.0.1
00000009A198   00000049DF98      0   GetVersionExA
00000009A1B6   00000049DFB6      0   GetTimeZoneInformation
00000009A2B8   00000049E0B8      0   GetFullPathNameA
00000009A2DA   00000049E0DA      0   GetExitCodeThread
00000004D698   00000044E298      0   JumpID("","%s")
000000006DFC   0000004079FC      0   MSH_WHEELSUPPORT_MSG
0000000855D4   0000004861D4      0   multipart/related; type="multipart/alternative"; boundary="=_NextPart_2relrfksadvnqindyw3nerasdf"
000000085640   000000486240      0   multipart/mixed; boundary="=_NextPart_2rfkindysadvnqw3nerasdf"
000000085688   000000486288      0   multipart/alternative; boundary="=_NextPart_2rfkindysadvnqw3nerasdf"
000000170832   000000577432      0   Smtp host:
000000070BD0   0000004717D0      0   System\CurrentControlSet\Control\Keyboard Layouts\%.8x
0000000951AA   000000495DAA      0   WebcamSettings1D
000000171650   000000578250      0           type="win32"
000000171666   000000578266      0           name="Microsoft.Windows.Common-Controls"
000000171698   000000578298      0           version="6.0.0.0"
0000001716B3   0000005782B3      0           publicKeyToken="6595b64144ccf1df"
0000001716DE   0000005782DE      0           language="*"
```

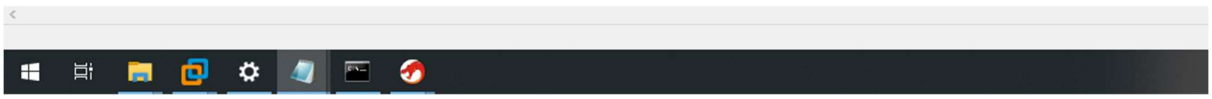


Figure:-1.3

Imports from "dependency walker.exe":

```
File  Edit  Format  View  Help
|kernnal32.dll                    |
|---------------------------------
|lstrcpyA                         |
|WriteFile                        |
|WaitForMultipleObjects           |
|VirtualQuery                     |
|VirtualAlloc                     |
|Sleep                            |
|ReleaseMutex                     |
|LeaveCriticalSection             |
|InitializeCriticalSection        |
|GetWindowsDirectoryA             |
|GetVersionExA                    |
|GetThreadLocale                  |
|GetTempPathA                     |
|FreeLibrary                      |
|FormatMessageA                   |
|DeleteCriticalSection            |
|CreateThread                     |
|CreateMutexA                     |
|CreateFileA                      |
---------------------------------------
GDI32.DLL
---------------------------------------|
SelectClipRgn
RealizePalette
GetClipBox
CreateBitmap
BitBlt
---------------------------------------
ADVAPI32.dll
---------------------------------------
RegQueryValueExA
RegOpenKeyExA
RegCloseKey
```

Figure:-1.4-a

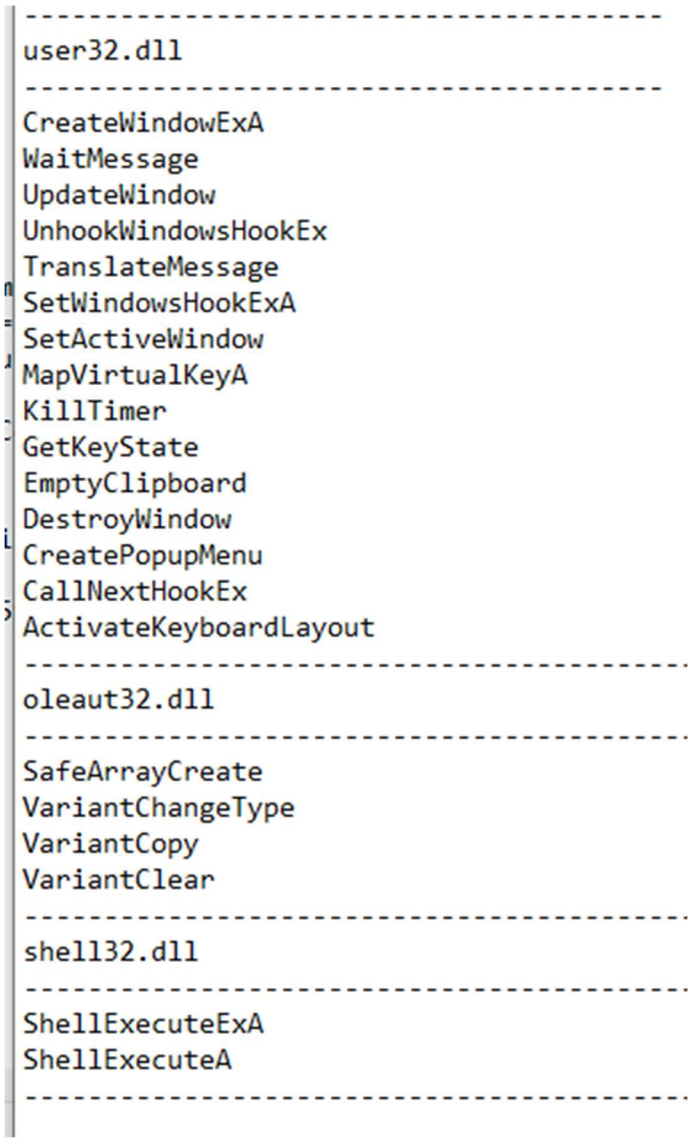


Figure:-1.4-b

The host-based indicators were:

1. pas.txt: This file was created when the credentials were entered in the windows live messenger. This file contained the credentials which were entered by the user.

2. Msnsettings.dat: It was created in C:\WINDOWS\msnsettings.dat location when the malware was run. This file was used as a configuration file by windowsLiveMessenger.exe.

3. C:\Program File\MSN Messenger\msnmsgr.exe: This could be one of the host base indicators as the path was mentioned inside this malware and in msnsettings.dat file.

Network-based indicators were:-

1. www.ourgodfather.com: Existing from windows live messenger application redirected to this URL through internet explorer browser.

2. yourpassword@password.com: The credentials stored in the pas.txt file might be forwarded to this email address.

3. The email addresses and URLs contents of msnsettings.dat.

Q-2. Describe how the malware works???

When windowslivemessenger.exe was opened, it created a msnsettings.dat file at location C:\WINDOWS\ automatically as shown in figure 2.1-a. This was also verified from the output of procmon.exe as shown in figure 2.1-b.

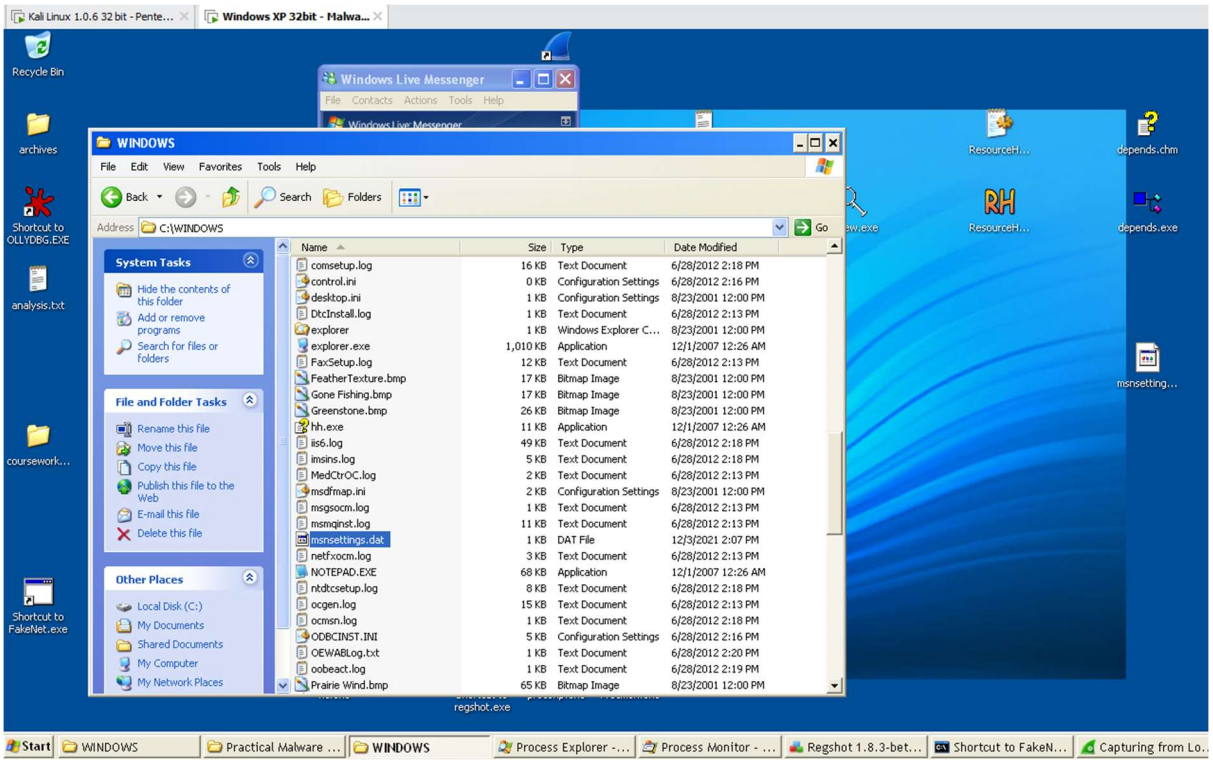


Figure:-2.1-a.

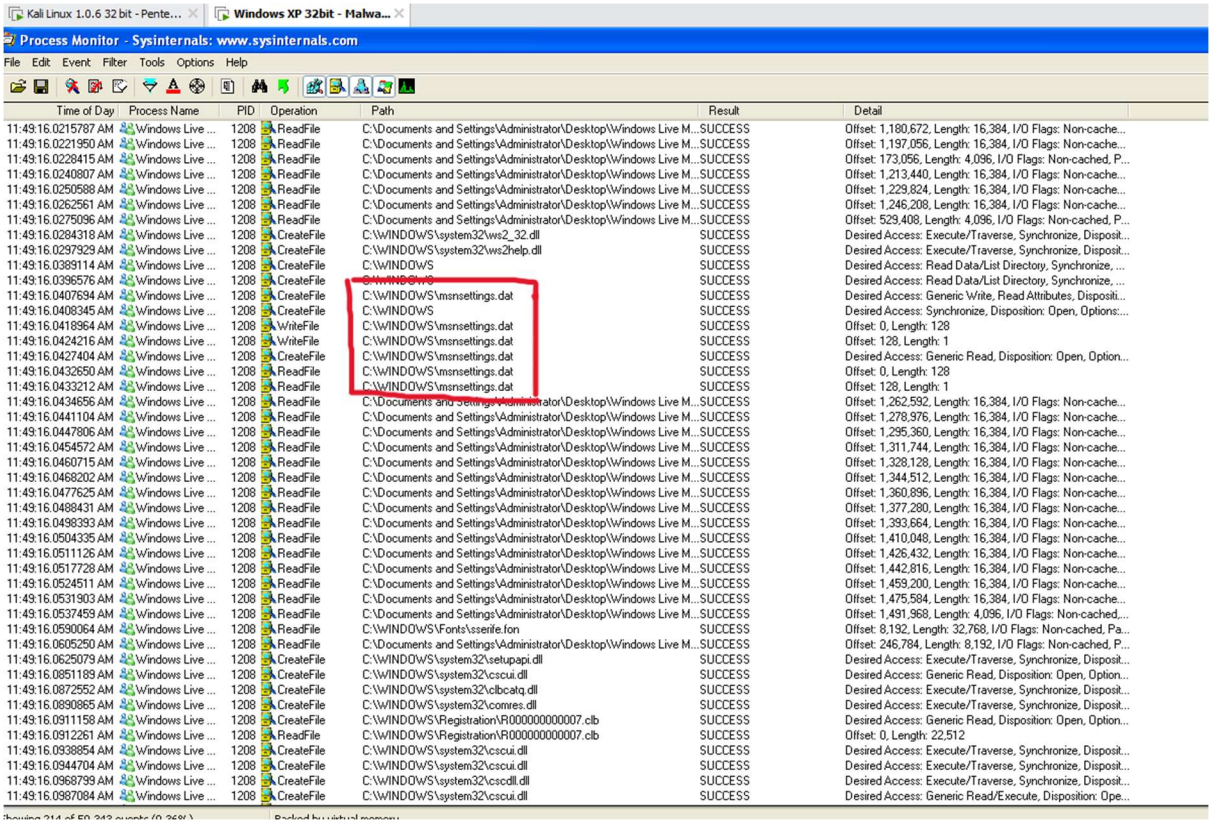


Figure:-2.1-b.

As the credentials were entered, the pas.txt file was created at location C:\, which contained the credentials provided by the user as shown in figures 2.2-a, 2.2-b, 2.2-c. The result of procmon.exe could be seen in figure 2.2-d.

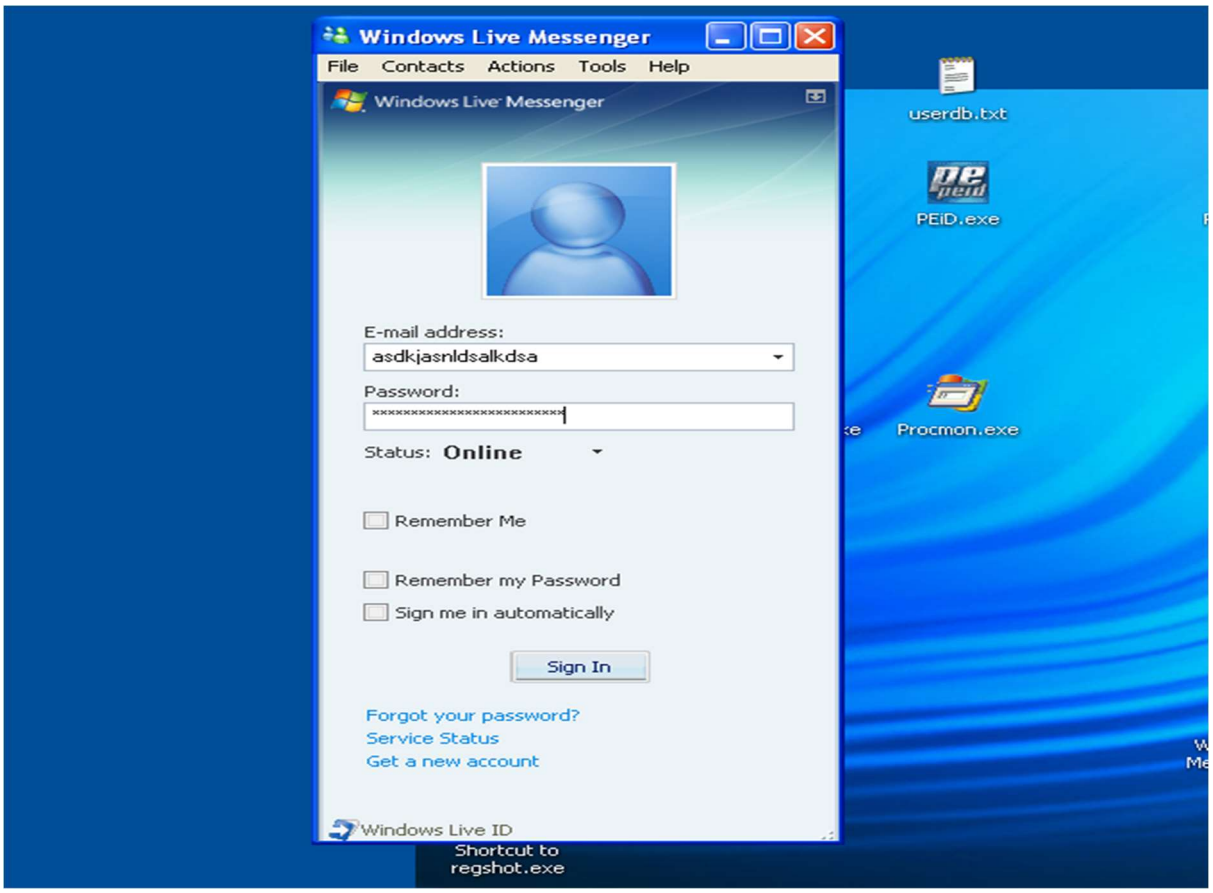


Figure:-2.2-a

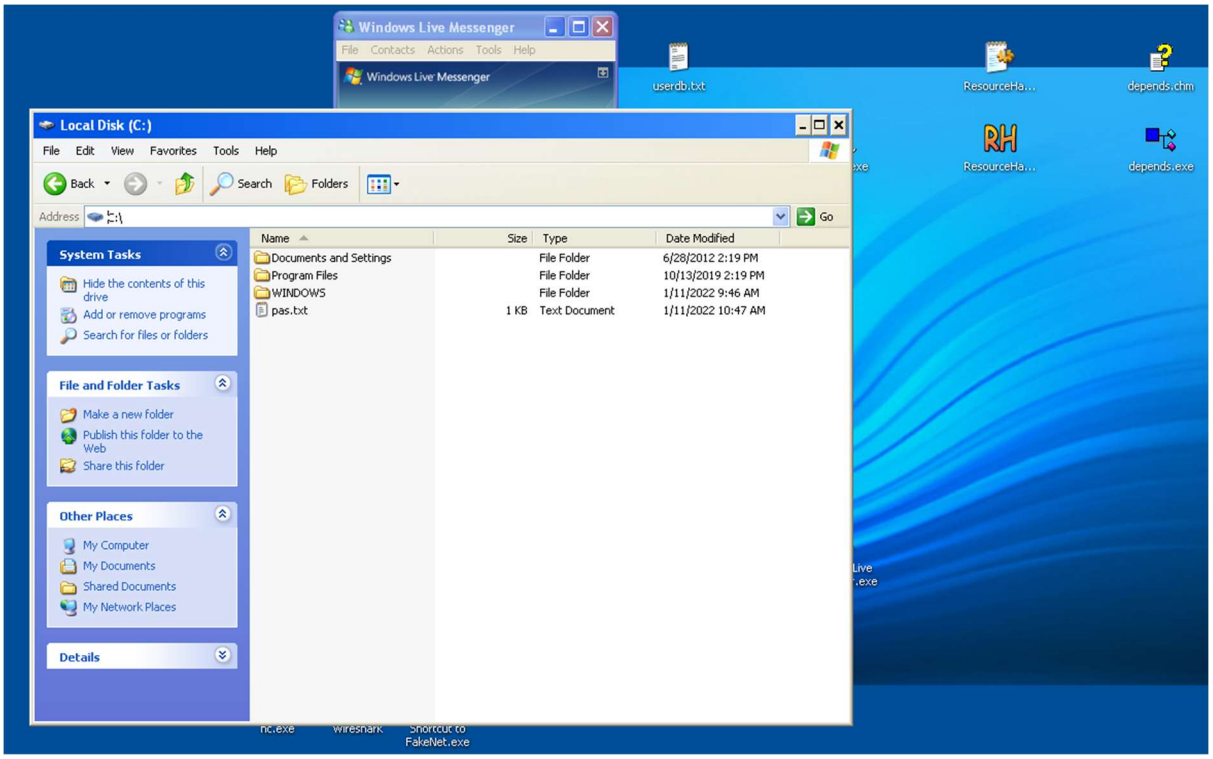


Figure:-2.2-b

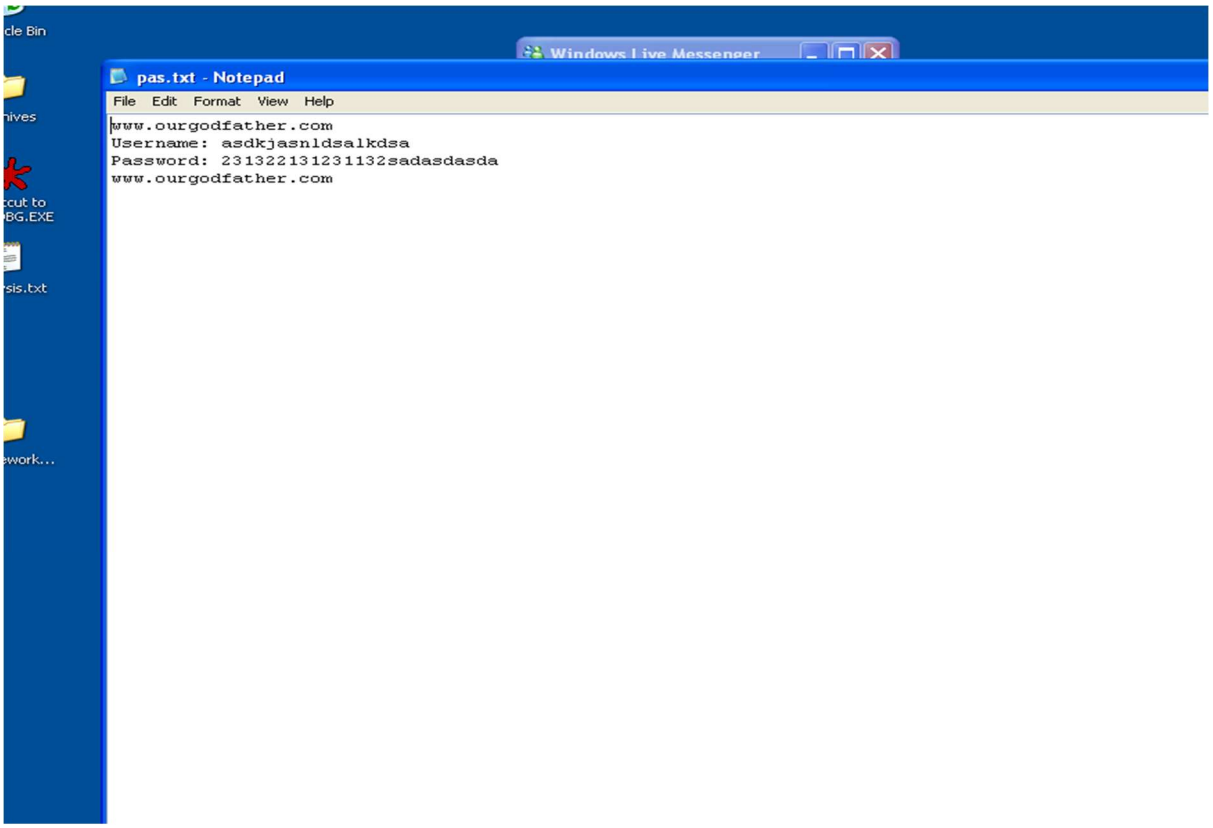


Figure:-2.2-c

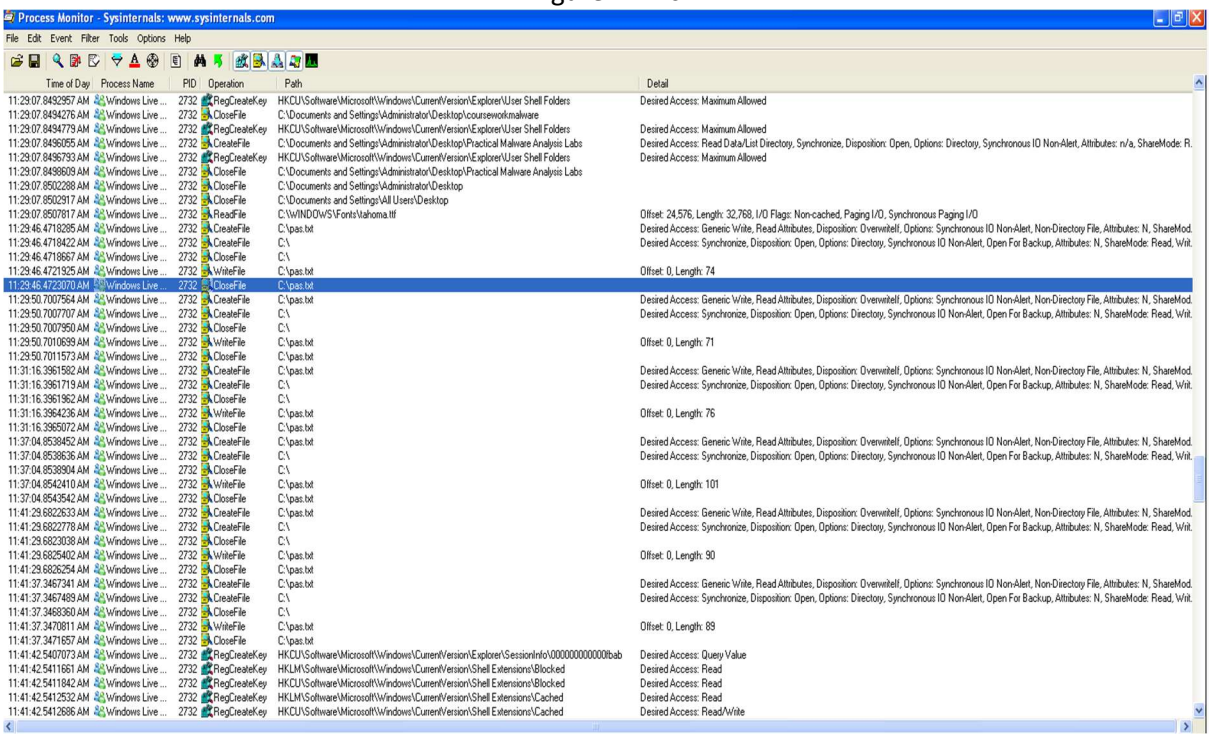


Figure:-2.2-d

Apart from all that, there were also a few values replaced by similar values with different INPUT which could be seen in the output provided by RegShot, as shown in Figure 2.3.

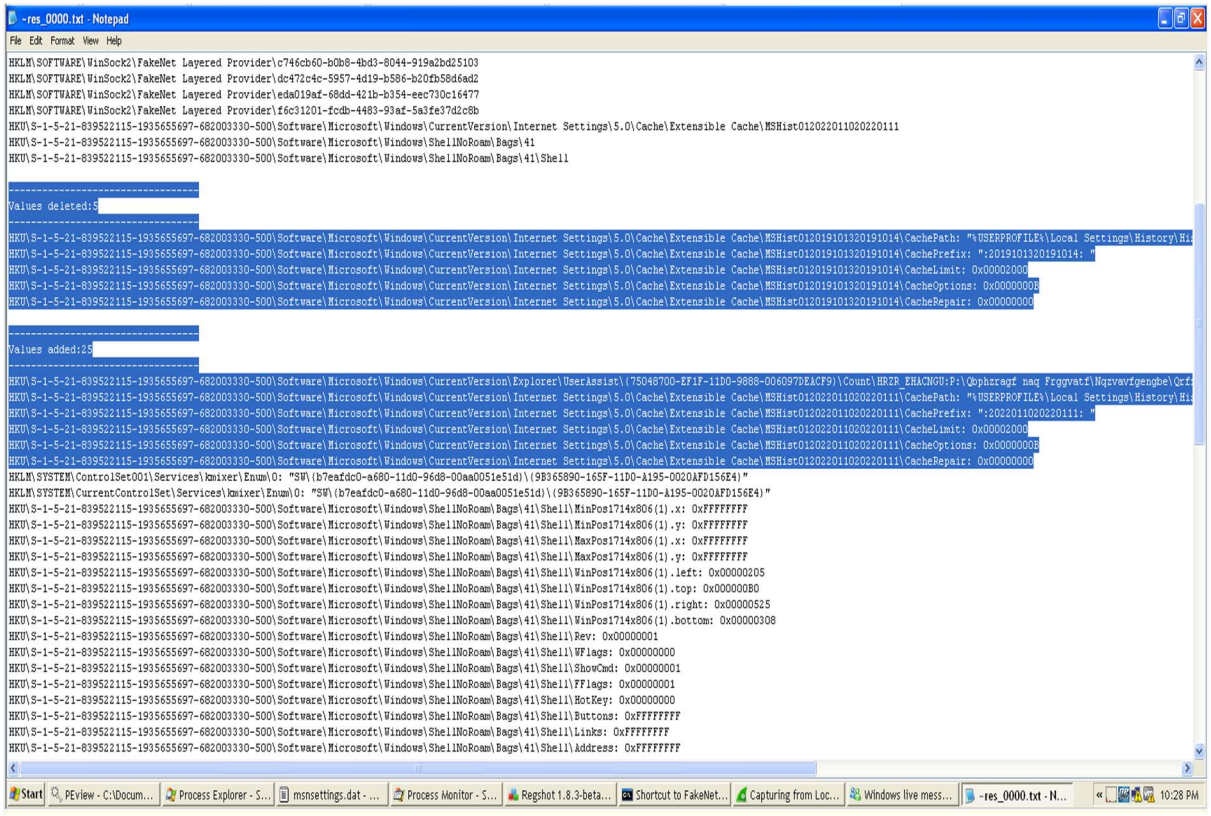


Figure:-2.3

The publisher of the application could be verified that it was not from an authenticated source, using the ProcessExplorer application as shown in figure 2.4. Hence the chances of it being malware were high. Also, mutexes generated by the malware could be observed in process explorer in figure 2.5.

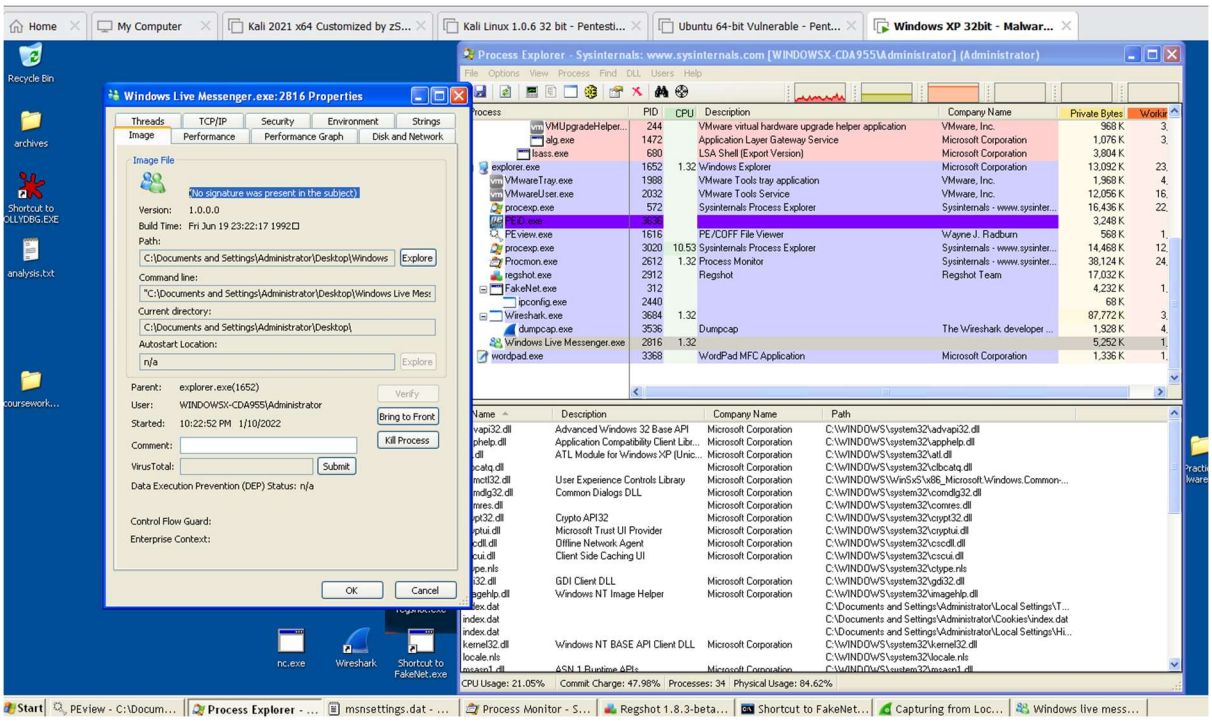


Figure:-2.4

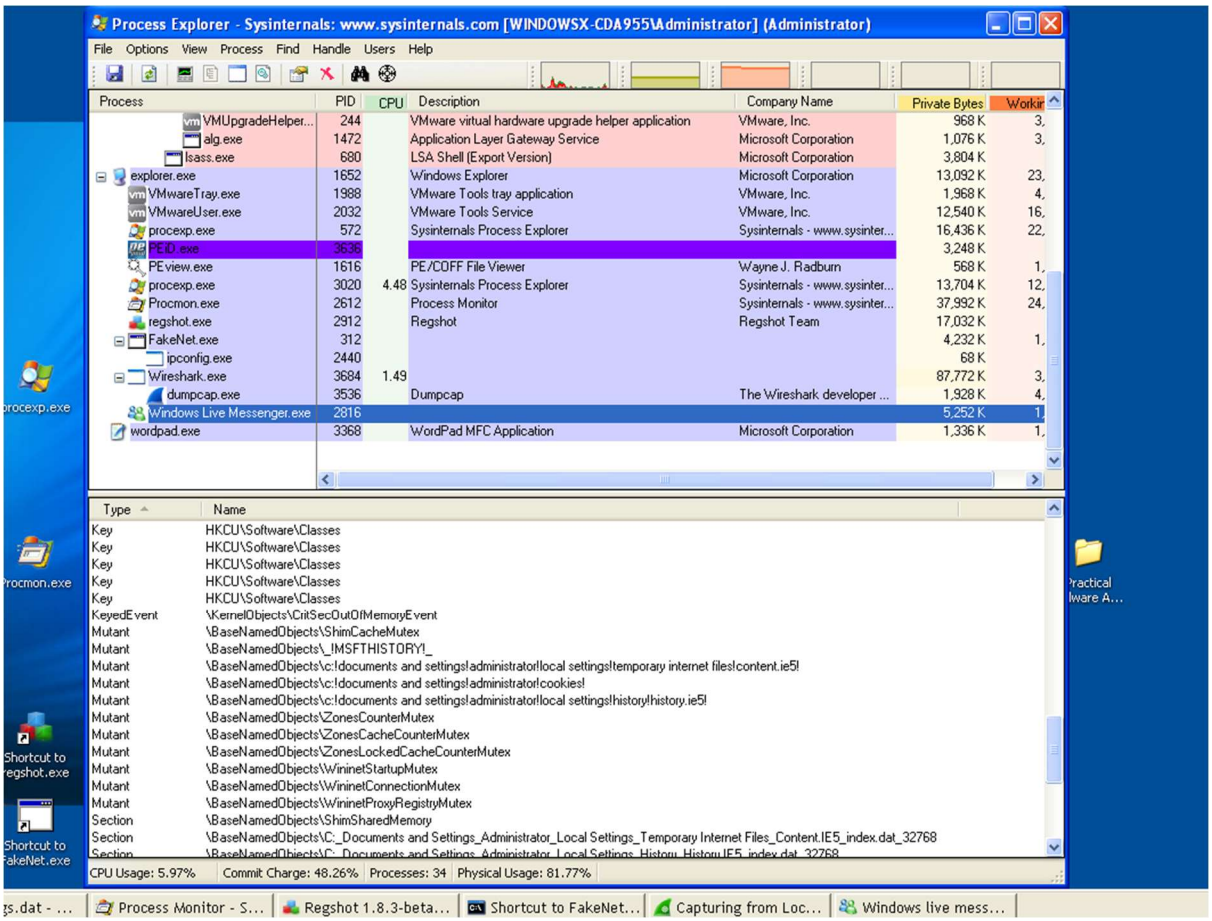


Figure:-2.5

Now, when the user entered the credentials and clicked on the sign-in, it could be observed that no significant packets were flowing through the network using Wireshark. However, if the generated msnsettings.dat file created at location C:\WINDOWS\ was replaced with the msnsettings.dat file provided in the course work folder, then it could be observed that the malware was trying to communicate with the website which was embedded in the msnsettings.dat file as shown in figure 2.6-a (two msnsetting.dat files) through the DNS port. This could be seen in the result provided by FakeNet.exe in figure 2.6-c.

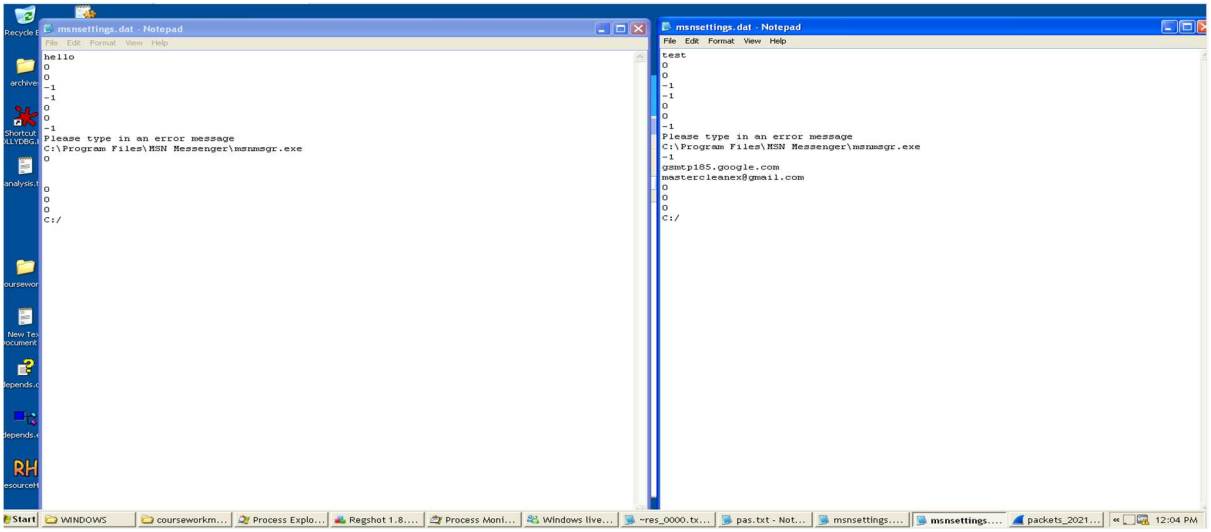


Figure:-2.6-a

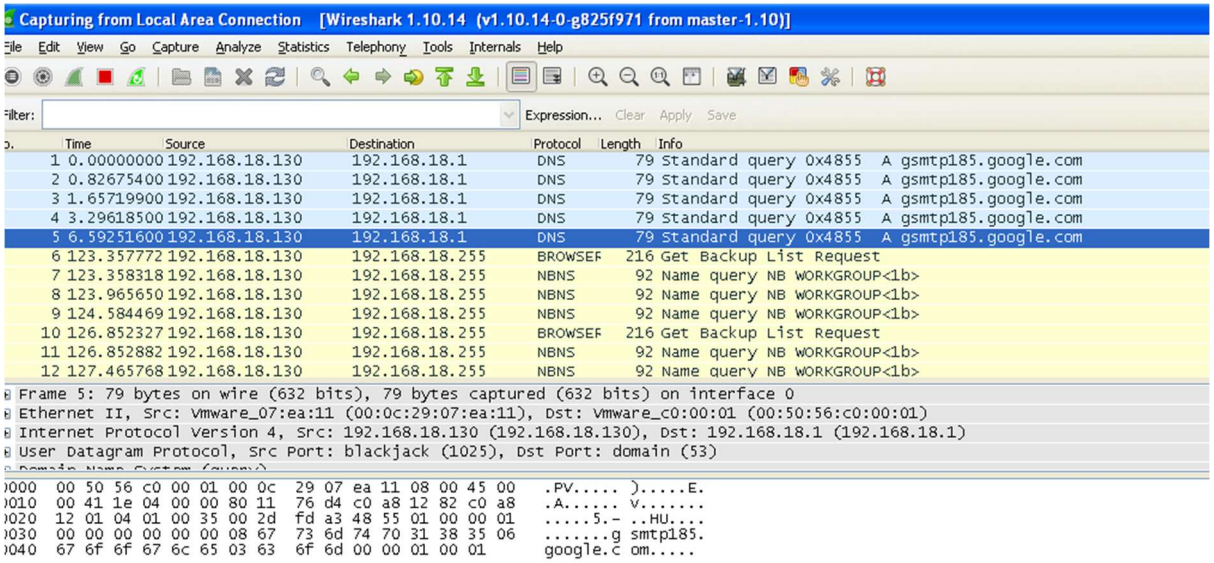


Figure:-2.6-b sending packets to gsmtp185.google.com.

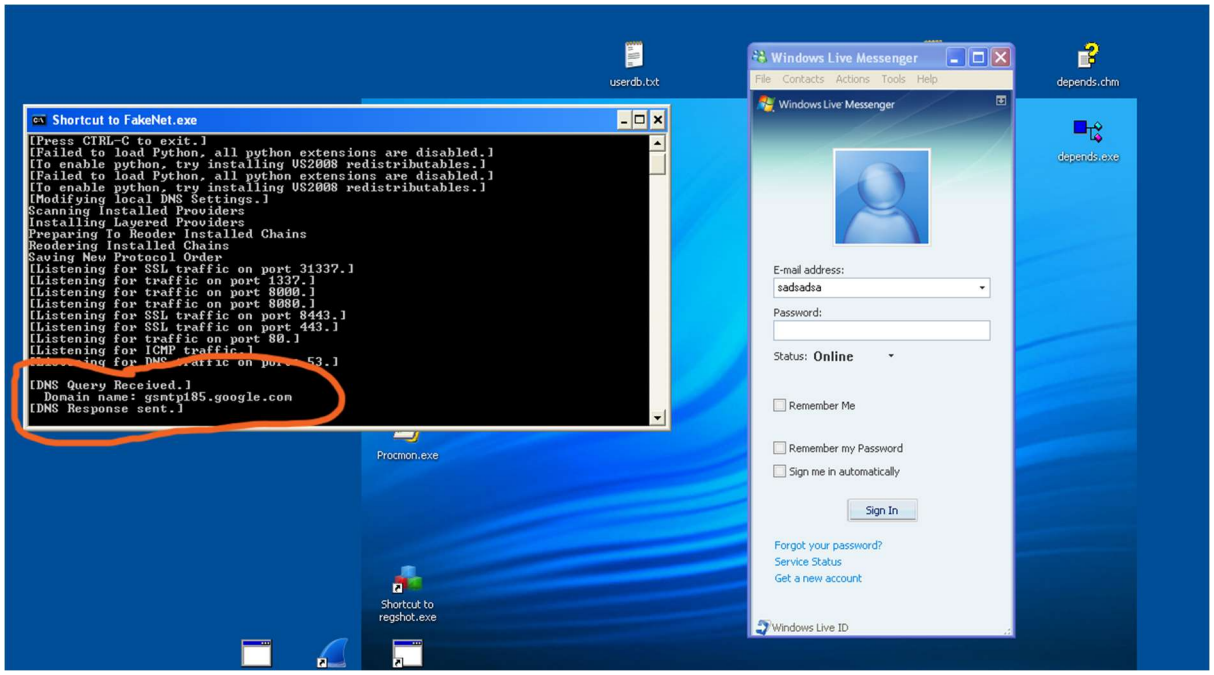


Figure:-2.6-c.

Another example in figure 2.7:-

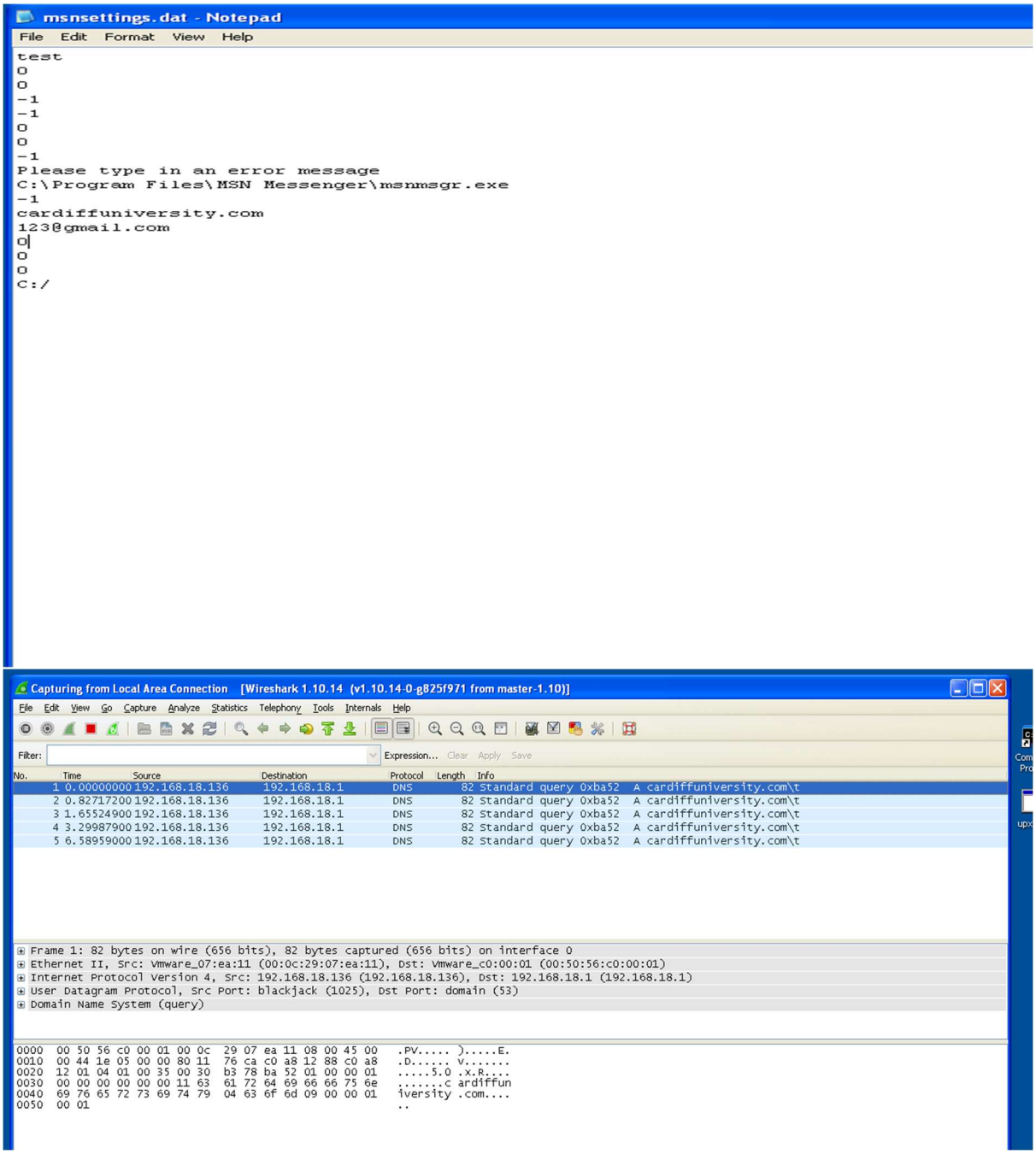


Figure:-2.7

As whatever the information was provided in the msnsetting.dat file, the malware would act based on it. Through all this information it was assumed that this file act as a configuration file.

Q-3. What the malware is trying to achieve???

The following observation was noticed when trying to access the windowslivemesserver.exe application:

- As it was proved that the malware was communicating with the website, embedded in the msnsettings.dat file, and credentials were stored in the pas.txt file, it could be assumed that the malware worked as a keylogger. Even though the malware was communicating with the website, there were no user credentials or a file(pas.txt) was observed flowing through the network in WireShark, as communication might be encrypted.
- It might be sending the user details through email, as in the msnsettings.dat file there was a column for the email address, and another email address yourpassword@password.com was observed from bin.TXT string.
- Whenever the windowslivemessenger.exe was closed, it redirected to www.ourgodfather.com, so there might be a possibility that the user credentials were transferred once the malware was terminated. This could be seen in the output provided by FakeNet.exe in figure 3.1

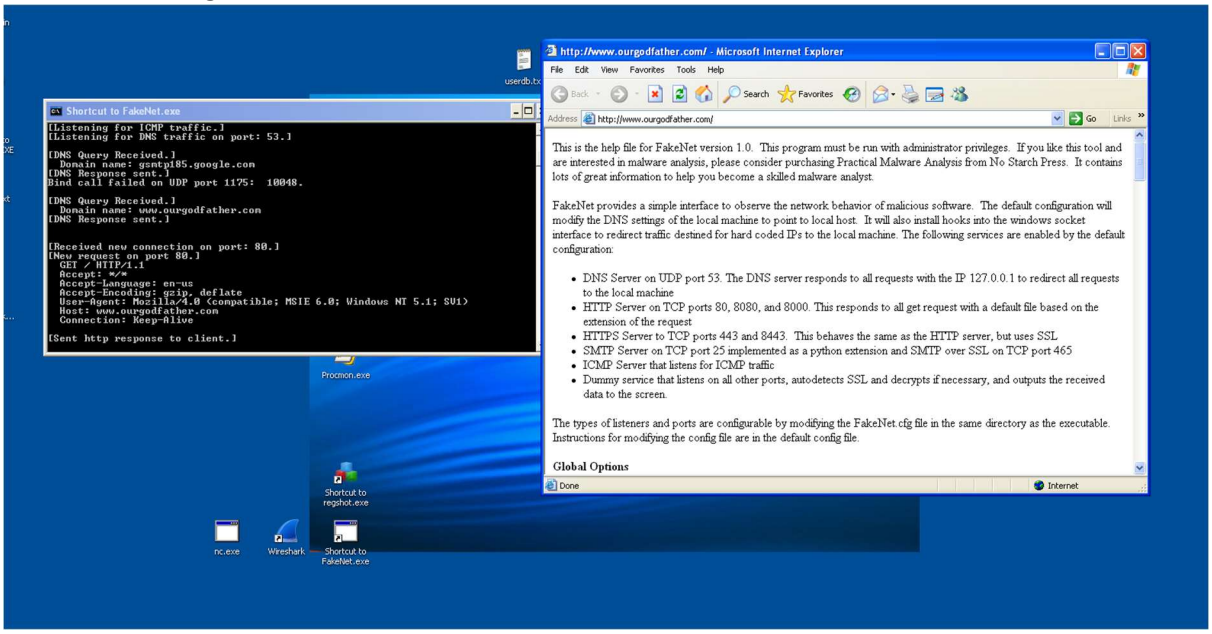


Figure:-3.1

Reverse engineering:

Q1.

The significant strings:-the heading of the four columns are Locations, String Values, String Representation, Data Types for figure 4.1-a,b.

.comment ::00000000	GC: (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0	"GC: (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0"	d s
004b6010	797d89ac982a50f680d8320ae90b63dee197fb9bd53390b6bdeabc4367e55943	"797d89ac982a50f680d8320ae90b63dee197fb9bd53390b6bdeabc4367e55943"	d s
004b6058	c5c65de8048e13ce87dbd4f76255026d0ac319de5df5123fa7ca8c93879f70ff1679f41718c586ef1b31700d1fb3a624	"c5c65de8048e13ce87dbd4f76255026d0ac319de5df5123fa7ca8c93879f70ff1679f41718c586ef1b31700d1fb3a624"	d s
004b60c4	[debug]: encryptin buffer;	"[debug]: encryptin buffer;"	d s
004b630a	6xeon_phi	"6xeon_phi"	d s
004b6314	Haswell	"haswell"	d s
004b6518	cannot set %fs base address for thread-local storage	"cannot set %fs base address for thread-local storage"	d s
004b6550	%s%s%s:%u: %s%sAssertion `%s' failed. %n	"%s%s%s:%u: %s%sAssertion `%s' failed.\n%n"	d s
004b6618	messages	"messages"	d s
004b6656	LC_MESSAGES	"LC_MESSAGES"	d s
004b6b6f	vfscanf-internal.c	"vfscanf-internal.c"	d s
004b6fa0	__vfscanf_internal	"__vfscanf_internal"	d s
004b6fc5	vfprintf-internal.c	"vfprintf-internal.c"	d s
004b7160	printf_positional	"printf_positional"	d s
004b7180	__vfprintf_internal	"__vfprintf_internal"	d s
004b73e0	0000000000000000	"0000000000000000"	d s
004b7770	malloc.c	"malloc.c"	d s
004b7804	corrupted double-linked list	"corrupted double-linked list"	d s
004b7903	malloc(): corrupted top size	"malloc(): corrupted top size"	d s
004b7920	realloc(): invalid old size	"realloc(): invalid old size"	d s
004b793c	!chunk_is_mmapped (oldp)	"!chunk_is_mmapped (oldp)"	d s
004b7972	malloc: top chunk is corrupt	"malloc: top chunk is corrupt"	d s
004b79da	system bytes = %10u	"system bytes = %10u\n"	d s
004b7a21	max mmap regions = %10u	"max mmap regions = %10u\n"	d s
004b7a70	%s%s%s:%u: %s%sAssertion `%s' failed.	"%s%s%s:%u: %s%sAssertion `%s' failed.\n"	d s
004b82b0	malloc(): memory corruption (fast)	"malloc(): memory corruption (fast)"	d s
004b8530	(unsigned long) (newsize) >= (unsigned long) (nb)	"(unsigned long) (newsize) >= (unsigned long) (nb)"	d s
004b85d0	!victim || chunk_is_mmapped (mem2chunk (victim)) || ar_ptr == arena_for_chunk (mem2chunk (victim))	"!victim || chunk_is_mmapped (mem2chunk (victim)) || ar_ptr == arena_for_chunk (mem2chunk (victim))"	d s

Figure:-4.1-a.

004b8638	!p || chunk_is_mmapped (mem2chunk (p)) || &main_arena == arena_for_chunk (mem2chunk (p))	"!p || chunk_is_mmapped (mem2chunk (p)) || &main_arena == arena_for_chunk (mem2chunk (p))"	d s
004b87a8	(char *) chunk2mem (p) + 4 * SIZE_SZ <= paligned_mem	"(char *) chunk2mem (p) + 4 * SIZE_SZ <= paligned_mem"	d s
004ba820	buffer overflow detected	"buffer overflow detected"	d s
004ba861	*** %s ***: terminated	"*** %s ***: terminated\n"	d s
004bb0c4	AVX512BW	"AVX512BW"	d s
004bb0cd	AVX512DQ	"AVX512DQ"	d s
004bb0d6	AVX512ER	"AVX512ER"	d s
004bb0df	AVX512PF	"AVX512PF"	d s
004bb0e8	AVX512VL	"AVX512VL"	d s
004bbbc0	__gconv_transform_internal_ucs2reverse	"__gconv_transform_internal_ucs2reverse"	d s
004ce7e0	0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ	"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"	d s
004cfc81	wrong ELF class: ELFCLASS32	"wrong ELF class: ELFCLASS32"	d s
004cfca0	only ET_DYN and ET_EXEC can be loaded	"only ET_DYN and ET_EXEC can be loaded"	d s
004cfcc8	ELF file version does not match current one	"ELF file version does not match current one"	d s
004d03a0	version->filename == NULL || ! _dl_name_match_p (version->filename, map)	"version->filename == NULL || ! _dl_name_match_p (version->filename, map)"	d s
004d03f0	symbol=%s; lookup in file=%s [%lu]	"symbol=%s; lookup in file=%s [%lu]\n"	d s
004d0418	marking %s [%lu] as NODELETE due to unique symbol	"marking %s [%lu] as NODELETE due to unique symbol\n"	d s
004d0450	version == NULL || !(flags & DL_LOOKUP_RETURN_NEWEST)	"version == NULL || !(flags & DL_LOOKUP_RETURN_NEWEST)"	d s
004d0c13	%s: cannot open file: %s	"%s: cannot open file: %s\n"	d s
004d0c2d	%s: cannot stat file: %s	"%s: cannot stat file: %s\n"	d s
004d0c80	%s: file is no correct profile data file for `%s'	"%s: file is no correct profile data file for `%s'\n"	d s
004d1085	DYNAMIC LINKER BUG!!!	"DYNAMIC LINKER BUG!!!"	d s
004d1732	Keld Simonsen	"Keld Simonsen"	d s
004d1740	keld@dkuug.dk	"keld@dkuug.dk"	d s
004d174e	+45 3122-6543	"+45 3122-6543"	d s
004d175c	+45 3325-6543	"+45 3325-6543"	d s
004d1772	1997-12-20	"1997-12-20"	d s
004d1780	ISO/IEC JTC1/SC22/WG20 - internationalization	"ISO/IEC JTC1/SC22/WG20 - internationalization"	d s
004d2a30	_dl_check_map_versions	"_dl_check_map_versions"	d s

Figure:-4.1-b

Imports:-

To reach the main function there is RDI:8 which leads to param_9 in the header of the entry function. Param_9 points to MainCaller as shown in figure 4.2.

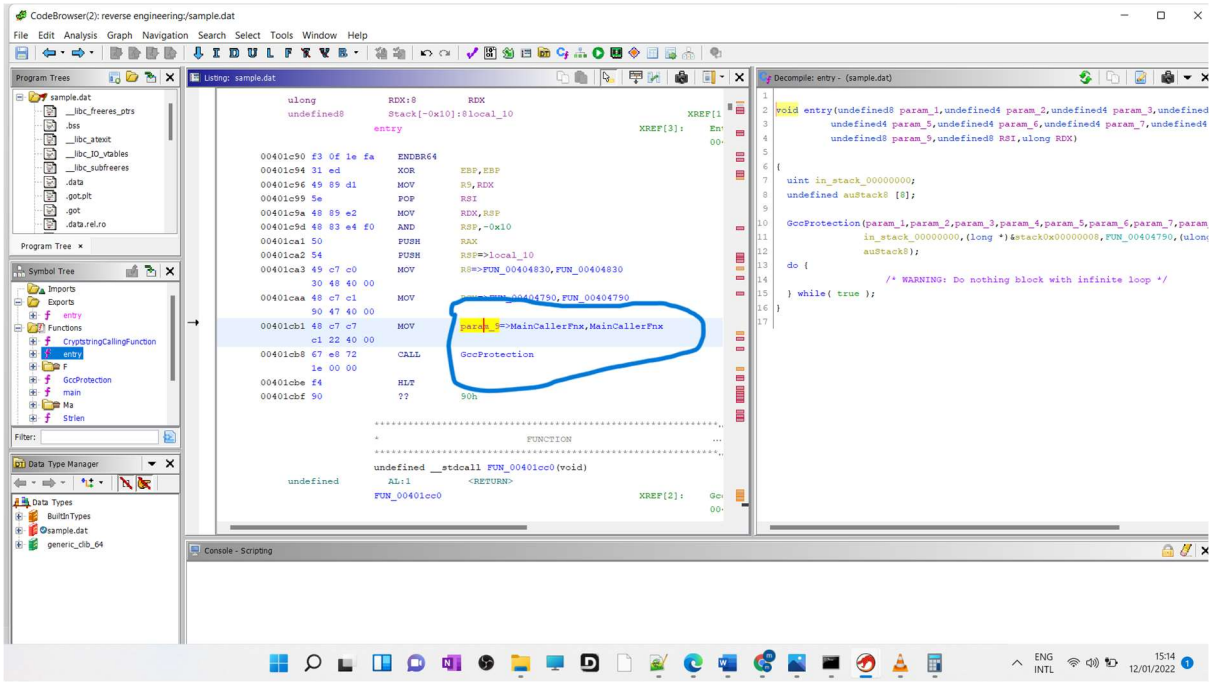


Figure:-4.2

The function which is called inside the MainCaller function is the main function as shown in figure 4.

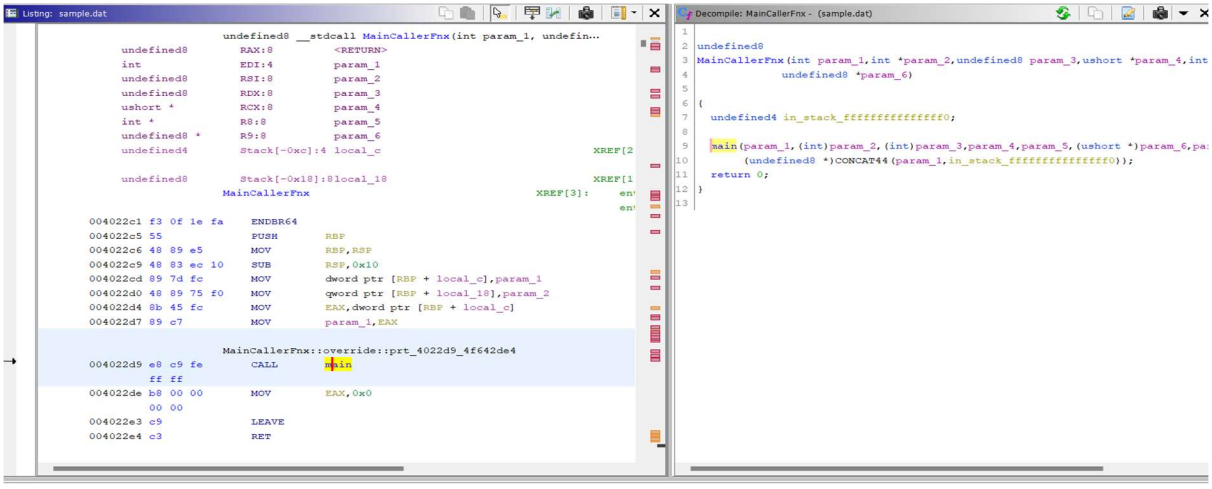


Figure:-4.3

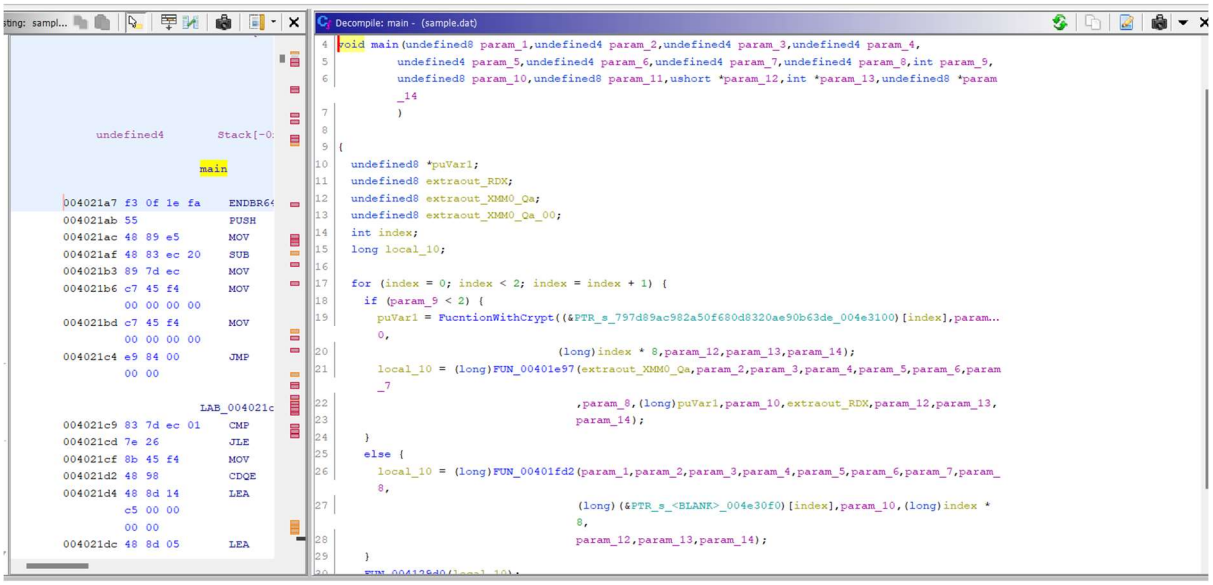


Figure:-4.4

Figure 4.4 shows the main function and FunctionWithCrypt which might be the encrypted string, and decryption is performed on it. Other functions which were discovered inside FunctionWithCrypt could be observed in figure 4.5-a, 4.5-b.

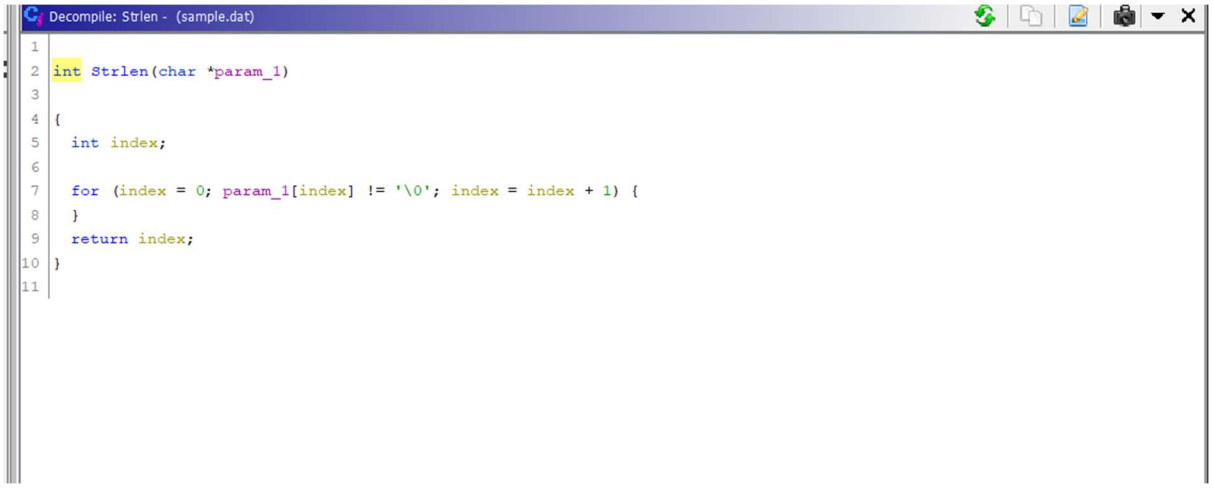
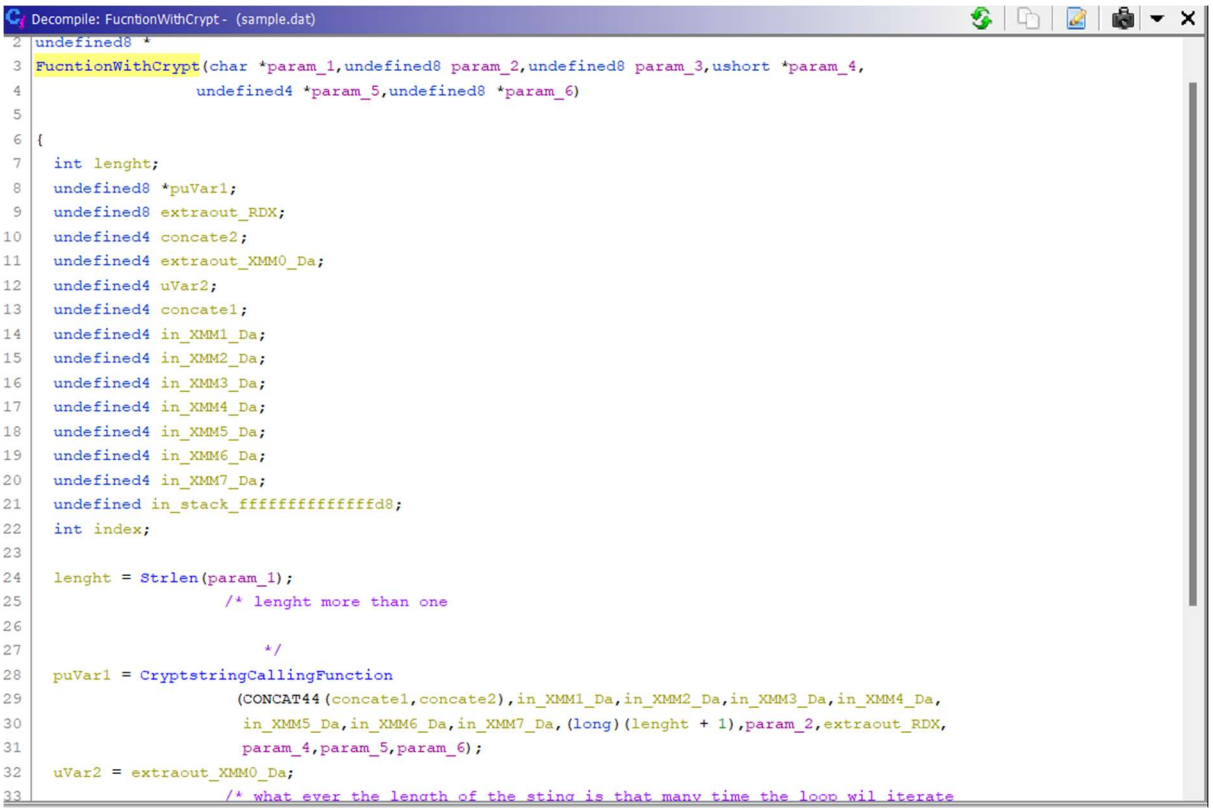


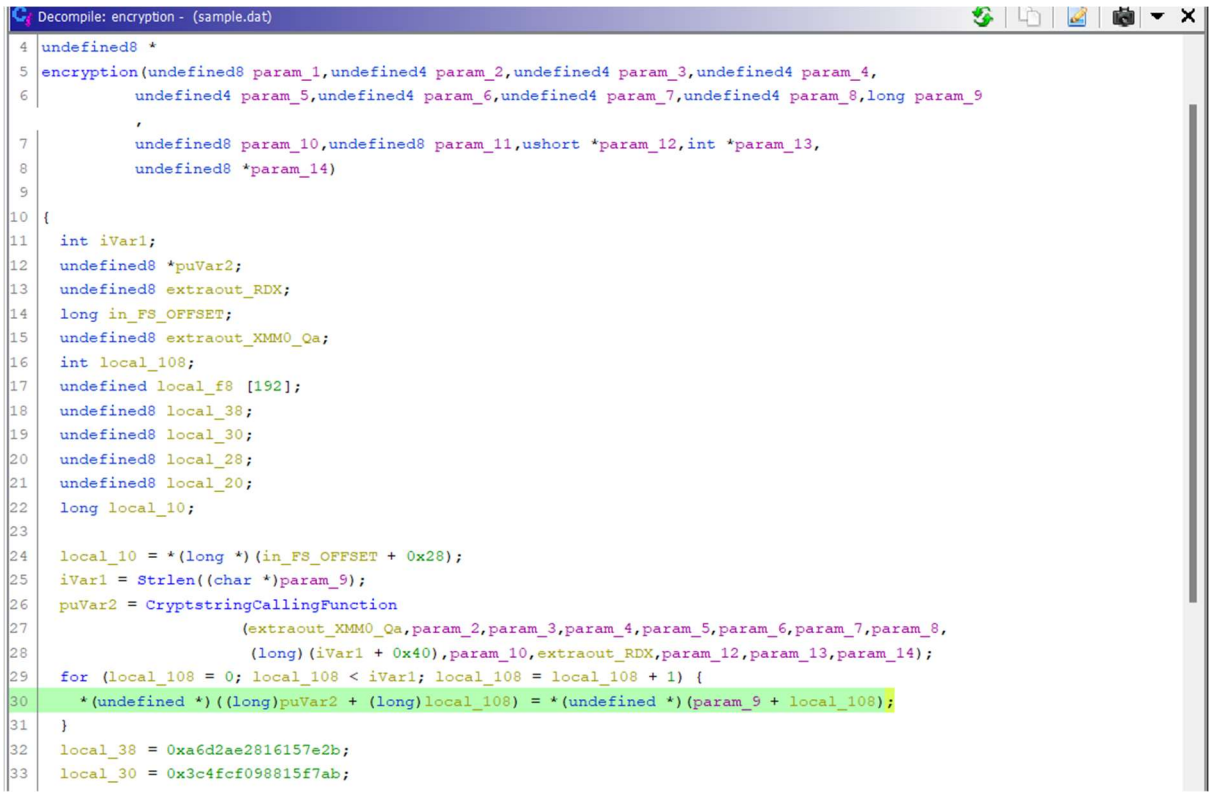
Figure:-4.5-a



```
Decompile: FucntionWithCrypt - (sample.dat)
undefined8 *
FucntionWithCrypt(char *param_1,undefined8 param_2,undefined8 param_3,ushort *param_4,
                  undefined4 *param_5,undefined8 *param_6)

{
  int lenght;
  undefined8 *puVar1;
  undefined8 extraout_RDX;
  undefined4 concate2;
  undefined4 extraout_XMM0_Da;
  undefined4 uVar2;
  undefined4 concate1;
  undefined4 in_XMM1_Da;
  undefined4 in_XMM2_Da;
  undefined4 in_XMM3_Da;
  undefined4 in_XMM4_Da;
  undefined4 in_XMM5_Da;
  undefined4 in_XMM6_Da;
  undefined4 in_XMM7_Da;
  undefined in_stack_ffffffffffffffd8;
  int index;

  lenght = Strlen(param_1);
                    /* lenght more than one

                       */
  puVar1 = CryptstringCallingFunction
                     (CONCAT44(concate1,concate2),in_XMM1_Da,in_XMM2_Da,in_XMM3_Da,in_XMM4_Da,
                      in_XMM5_Da,in_XMM6_Da,in_XMM7_Da,(long)(lenght + 1),param_2,extraout_RDX,
                      param_4,param_5,param_6);
  uVar2 = extraout_XMM0_Da;
                    /* what ever the length of the sting is that many time the loop wil iterate
```

Figure:-4.5-b.

The function FUN_00401e97 which is renamed as encryption could be seen in figure 4.6. this might contain all the encryption.



```
Decompile: encryption - (sample.dat)
undefined8 *
encryption(undefined8 param_1,undefined4 param_2,undefined4 param_3,undefined4 param_4,
           undefined4 param_5,undefined4 param_6,undefined4 param_7,undefined4 param_8,long param_9
           ,
           undefined8 param_10,undefined8 param_11,ushort *param_12,int *param_13,
           undefined8 *param_14)

{
  int iVar1;
  undefined8 *puVar2;
  undefined8 extraout_RDX;
  long in_FS_OFFSET;
  undefined8 extraout_XMM0_Qa;
  int local_108;
  undefined local_f8 [192];
  undefined8 local_38;
  undefined8 local_30;
  undefined8 local_28;
  undefined8 local_20;
  long local_10;

  local_10 = *(long *)(in_FS_OFFSET + 0x28);
  iVar1 = Strlen((char *)param_9);
  puVar2 = CryptstringCallingFunction
                     (extraout_XMM0_Qa,param_2,param_3,param_4,param_5,param_6,param_7,param_8,
                      (long)(iVar1 + 0x40),param_10,extraout_RDX,param_12,param_13,param_14);
  for (local_108 = 0; local_108 < iVar1; local_108 = local_108 + 1) {
    *(undefined *)((long)puVar2 + (long)local_108) = *(undefined *)(param_9 + local_108);
  }
  local_38 = 0xa6d2ae2816157e2b;
  local_30 = 0x3c4fcf098815f7ab;
```

Figure:-4.6

Q-2

AES encryption contains s-box values, r-box values, and r-con values as shown in Figures 5.1-a and b.

AES S-box

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

The column is determined by the least significant nibble, and the row by the most significant nibble. For example, the value $9a_{16}$ is converted into $b8_{16}$.

Figure:-5.1

	1	2	3	4	5	6	7	8	9	0a	0b	0c	0d	0e	0f	
0	52	9	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
10	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
20	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
30	8	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
40	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
50	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
60	90	d8	ab	0	8c	bc	d3	0a	f7	e4	58	5	b8	b3	45	6
70	d0	2c	1e	8f	ca	3f	0f	2	c1	af	bd	3	1	13	8a	6b
80	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
90	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a0	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b0	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c0	1f	dd	a8	33	88	7	c7	31	b1	12	10	59	27	80	ec	5f
d0	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e0	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f0	17	2b	4	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Figure:-5.1-rs-box

The similar values can be searched and observed in the ghidra as shown in figure 5.2-a, b, c

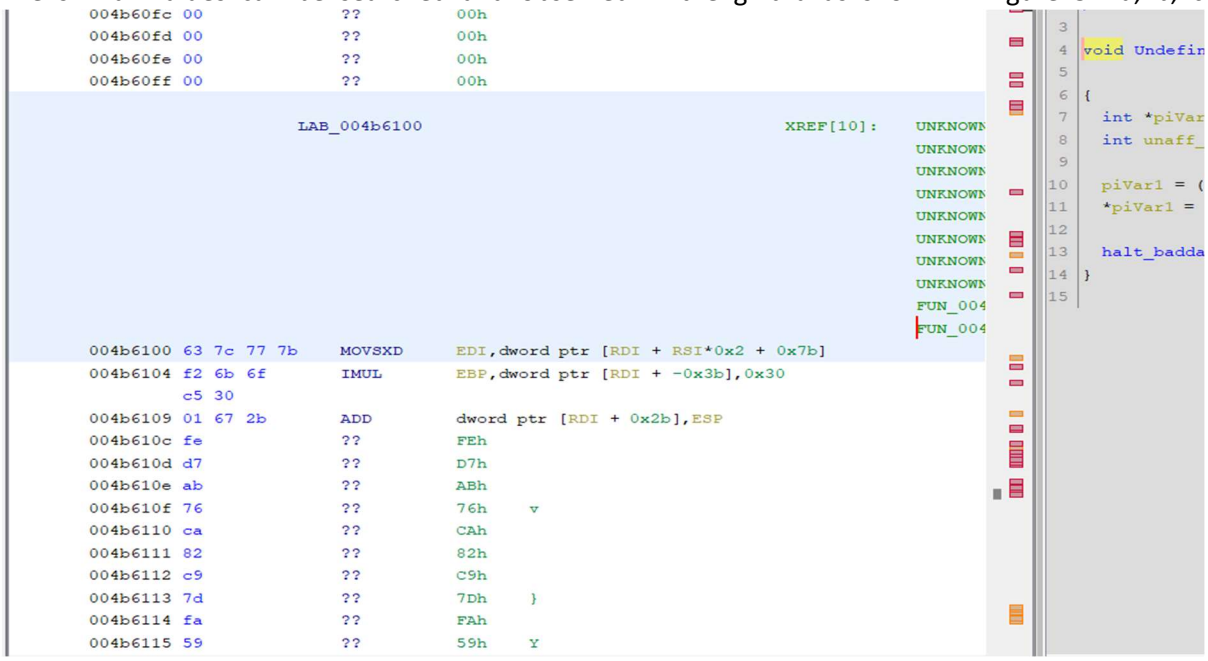


Figure:-5.2-a s-box

```
                    DAT_004b6200                                    XREF[2]:     FU
                                                                                 FU
004b6200 52             ??          52h    R
004b6201 09             ??          09h
004b6202 6a             ??          6Ah    j
004b6203 d5             ??          D5h
004b6204 30             ??          30h    0
004b6205 36             ??          36h    6
004b6206 a5             ??          A5h
004b6207 38             ??          38h    8
004b6208 bf             ??          BFh
004b6209 40             ??          40h    @
004b620a a3             ??          A3h
004b620b 9e             ??          9Eh
004b620c 81             ??          81h
004b620d f3             ??          F3h
004b620e d7             ??          D7h
004b620f fb             ??          FBh
004b6210 7c             ??          7Ch    |
004b6211 e3             ??          E3h
004b6212 39             ??          39h    9
```

Figure:-5.2-b rs-box

```
                    DAT_004b6300                                    XREF[1]:     UNKNOWI
004b6300 8d             ??          8Dh

                    DAT_004b6301                                    XREF[1]:     UNKNOWI
004b6301 01             undefine…   01h
004b6302 02             ??          02h
004b6303 04             ??          04h
004b6304 08             ??          08h
004b6305 10             ??          10h
004b6306 20             ??          20h
004b6307 40             ??          40h    @
004b6308 80             ??          80h
004b6309 1b             ??          1Bh
                    s_xeon_phi_004b630b                             XREF[0,9]:   GccProt
                    s_i_004b6312                                                 KeyRela
                                                                                 KeyRela
                                                                                 KeyRela
                                                                                 KeyRela
```

Figure:-5.2-c r-con

Next to s-box, there are functions in figure 5.3:-

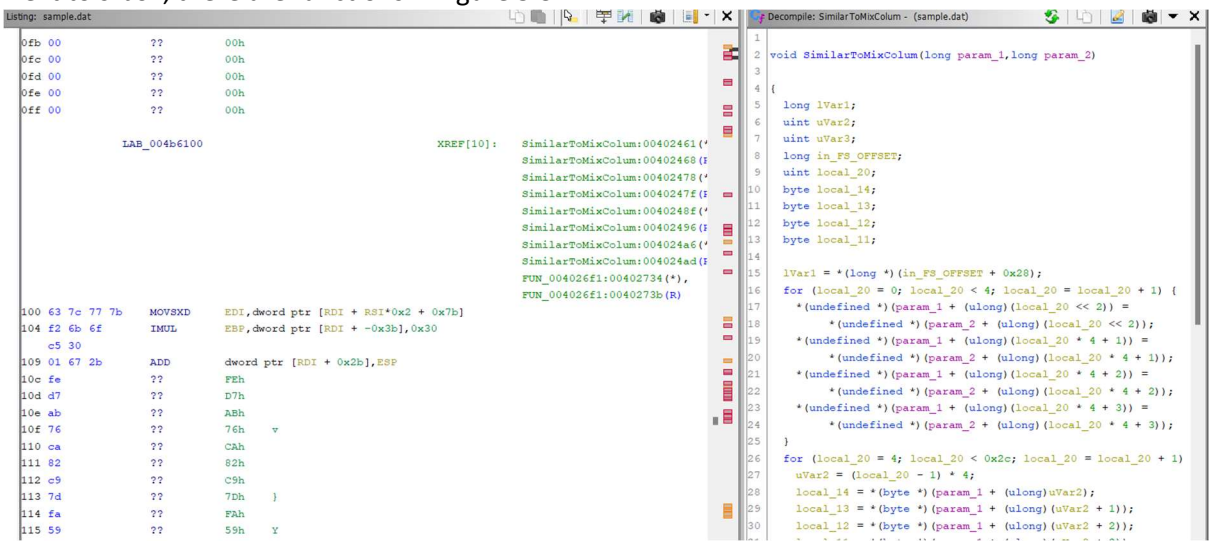


Figure:-5.3

It is assumed that this is the same function as that in the AES encryption which is MixColumn.

Moreover, according to me, the key should be in the function which is shown in figure 5.4.

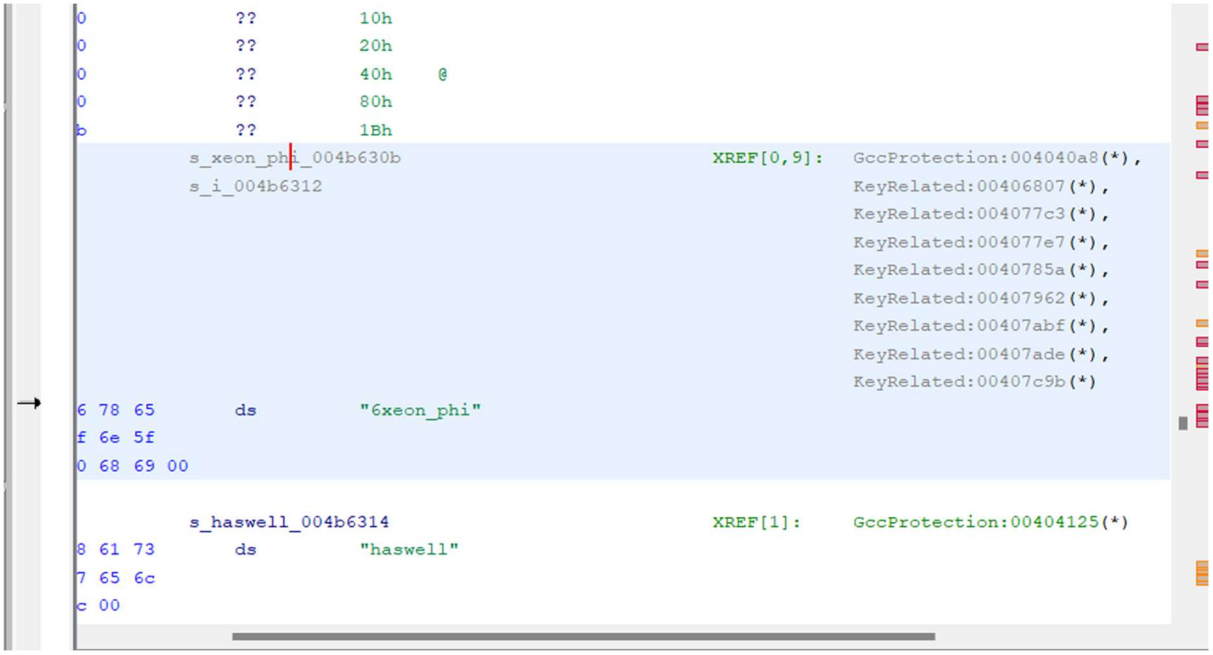
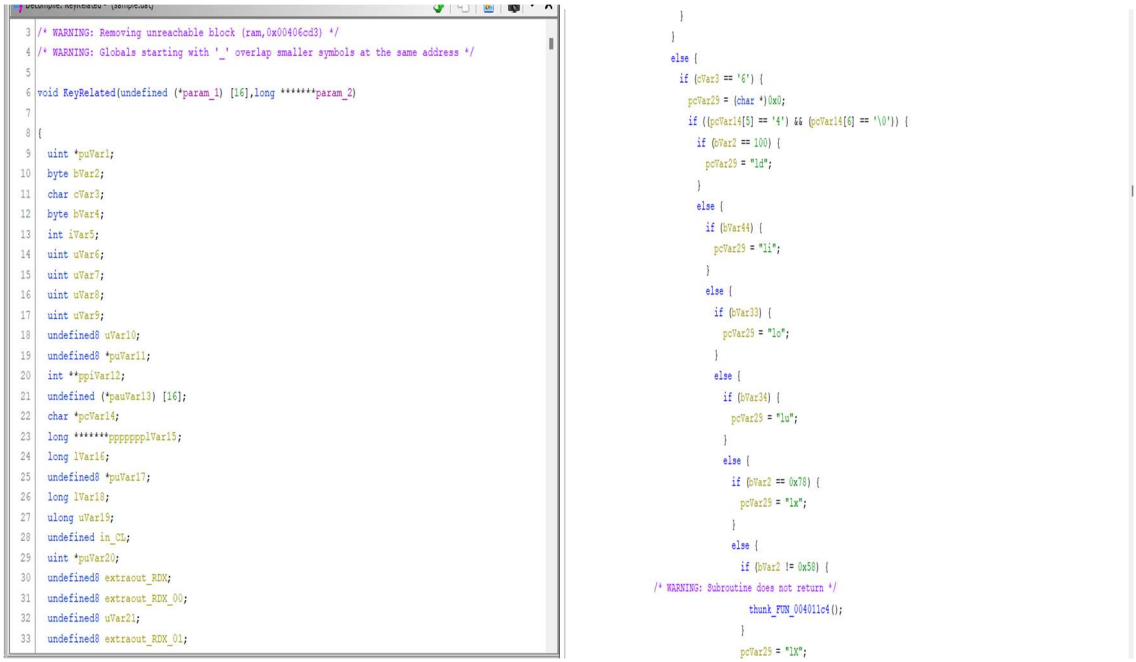


Figure:-5.4-a



```
/* WARNING: Removing unreachable block (ram,0x00406cd3) */
/* WARNING: Globals starting with '_' overlap smaller symbols at the same address */

void KeyRelated(undefined (*param_1) [16],long *******param_2)

{
  uint *puVar1;
  byte bVar2;
  char cVar3;
  byte bVar4;
  int iVar5;
  uint uVar6;
  uint uVar7;
  uint uVar8;
  uint uVar9;
  undefined8 uVar10;
  undefined8 *puVar11;
  int **ppiVar12;
  undefined (*pauVar13) [16];
  char *pcVar14;
  long *******ppppppplVar15;
  long lVar16;
  undefined8 *puVar17;
  long lVar18;
  ulong uVar19;
  undefined in_CL;
  uint *puVar20;
  undefined8 extraout_RDX;
  undefined8 extraout_RDX_00;
  undefined8 uVar21;
  undefined8 extraout_RDX_01;
```

```
    }
  }
  else {
    if (cVar3 == 'G') {
      pcVar29 = (char *)0x0;
      if ((pcVar14[5] == '4') && (pcVar14[6] == '\0')) {
        if (bVar2 == 100) {
          pcVar29 = "ld";
        }
        else {
          if (bVar44) {
            pcVar29 = "li";
          }
          else {
            if (bVar33) {
              pcVar29 = "lo";
            }
            else {
              if (bVar34) {
                pcVar29 = "lu";
              }
              else {
                if (bVar2 == 0x78) {
                  pcVar29 = "lx";
                }
                else {
                  if (bVar2 != 0x58) {
/* WARNING: Subroutine does not return */
                    thunk_FUN_004011c4();
                  }
                  pcVar29 = "lX";
```

Figure:-5.4-b

Q3.

The result from Virus Total states there is no malicious flags were detected as this can be seen in figure 6.1.

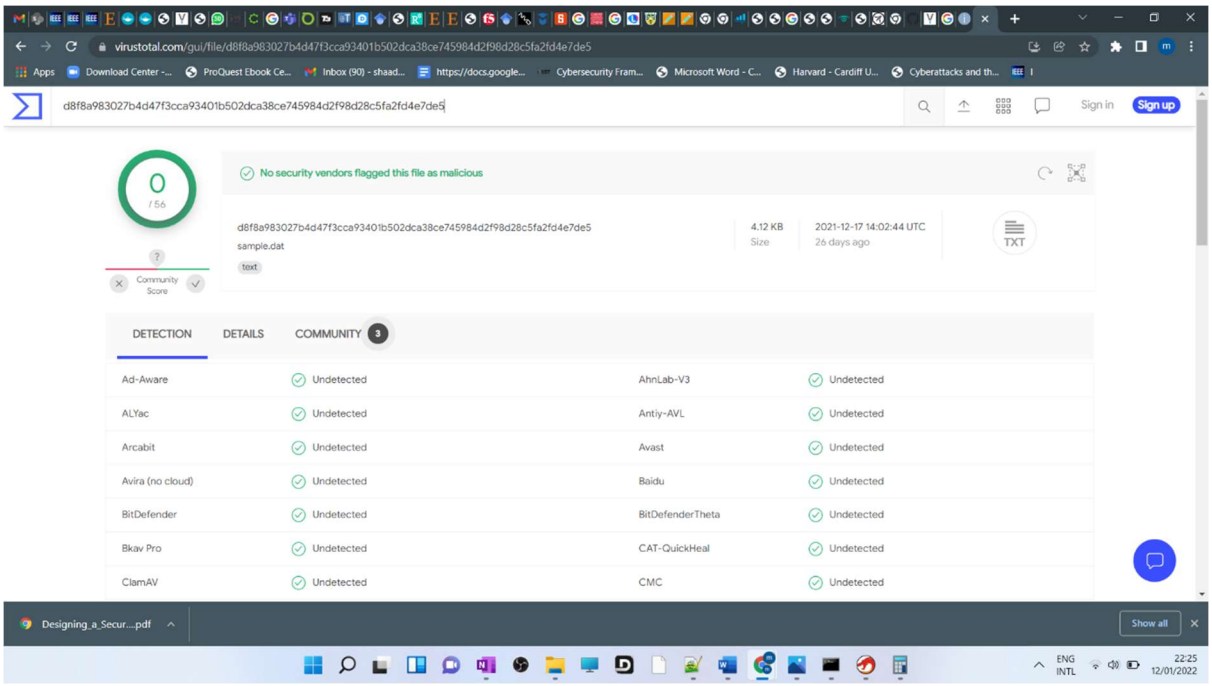


Figure:-6.1

By keeping this in mind it could be concluded that there used to be an encryption and decryption function inside this malware but cannot accept any inputs due to which it cannot corrupt any files.