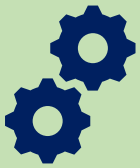
REVERSE ENGINEERING

Prof. Adel KHALDI



By,

Bharani MOORTHY M.E SNS

Nikhil NAGARAJAN M.E SNS

INDEX

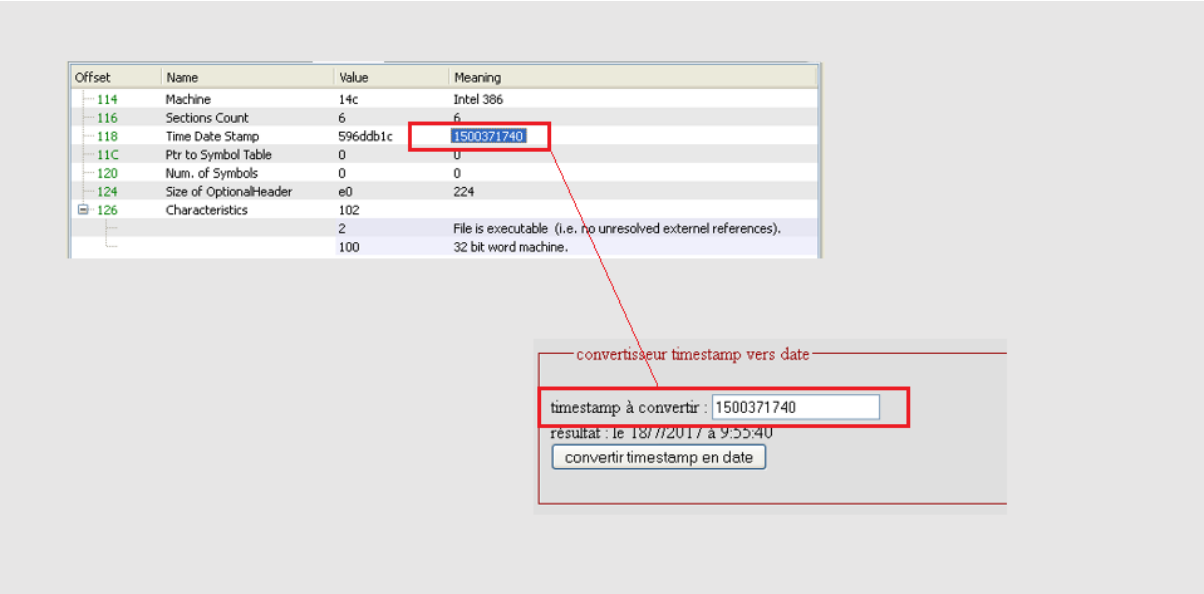
Exam_part_01.html

1.Which compiler was used?

File type : .exe (Executable) – Intel 80386
This was done using linux command through file command and PE Bear.

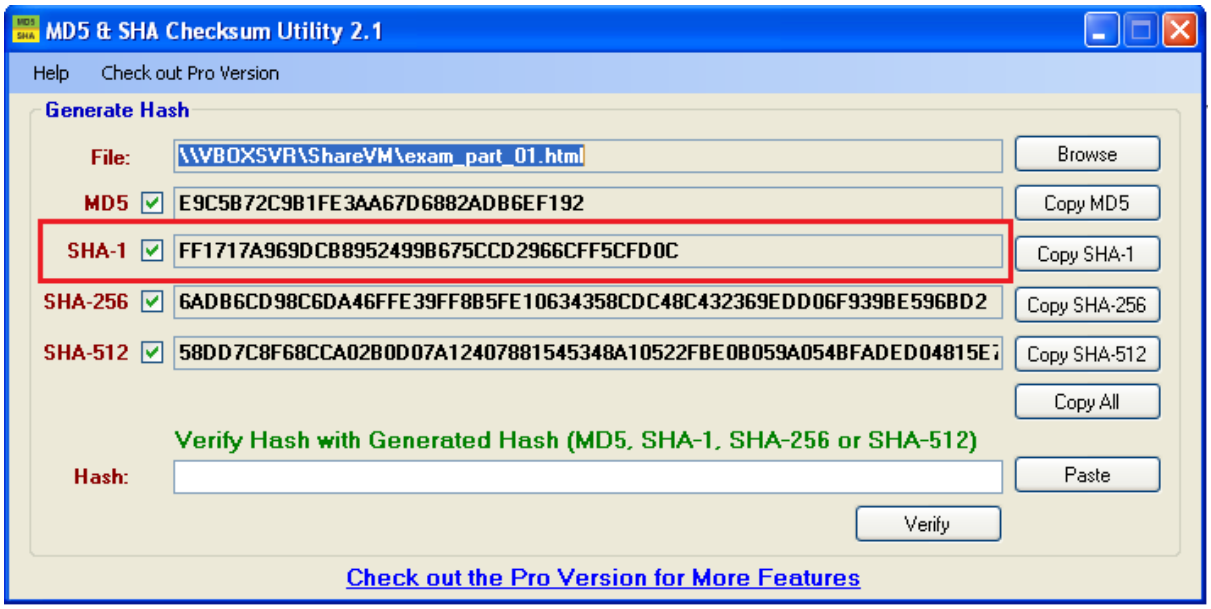
```
root@box:/media/sf_ShareVM# file exam_part_01.html
exam_part_01.html: PE32 executable (GUI) Intel 80386, for MS Windows
```

2.When was this application compiled?



Timestamp: Tuesday, July 18, 2017 09:55:40 AM GMT+02:00 DST
This was followed through under file header options in PE Bear.

3.What is its SHA-1 hash value?

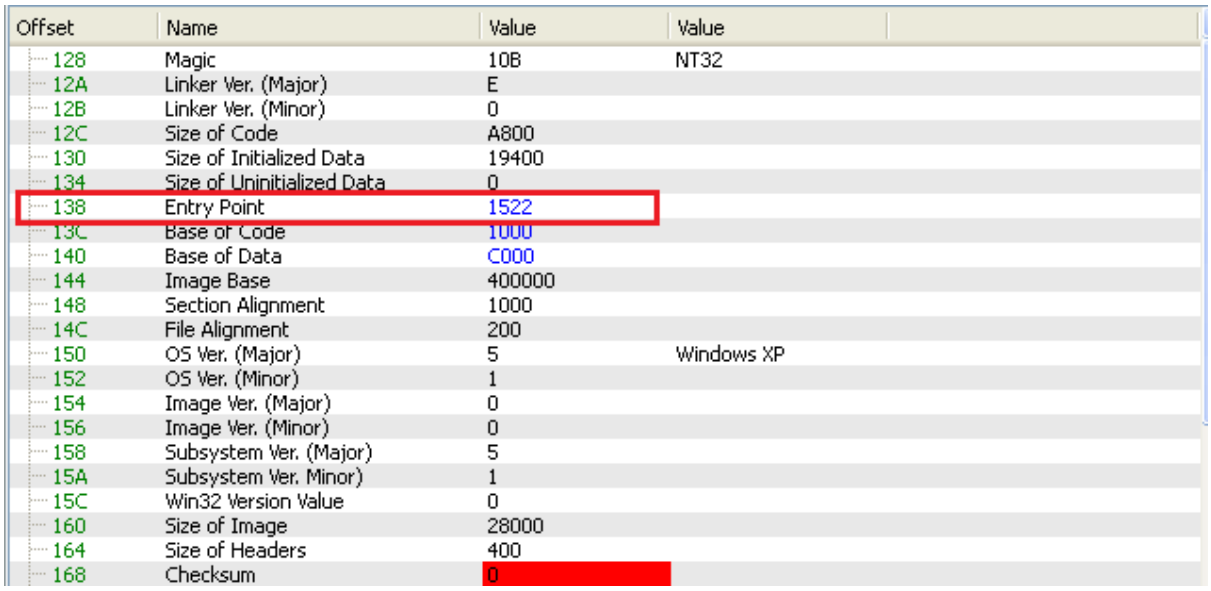


File Hash (SHA-1):ff1717a969dcb8952499b675ccd2966cff5cfd0c
This was done via MD5 and SHA checksum utility

4.Which CPU platform is it compiled for? 32- or 64-bit versions?

Its compiled-on Intel 80386 and It is 32- bit version.

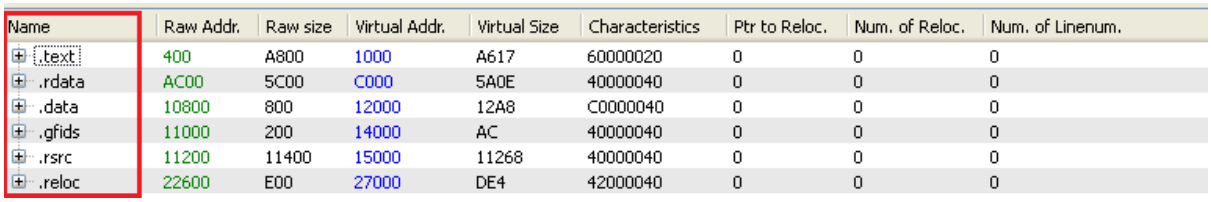
5.What it its entry point? In which section is it?



Offset	Name	Value	Value
128	Magic	10B	NT32
12A	Linker Ver. (Major)	E	
12B	Linker Ver. (Minor)	0	
12C	Size of Code	A800	
130	Size of Initialized Data	19400	
134	Size of Uninitialized Data	0	
138	Entry Point	1522	
13C	Base of Code	1000	
140	Base of Data	C000	
144	Image Base	400000	
148	Section Alignment	1000	
14C	File Alignment	200	
150	OS Ver. (Major)	5	Windows XP
152	OS Ver. (Minor)	1	
154	Image Ver. (Major)	0	
156	Image Ver. (Minor)	0	
158	Subsystem Ver. (Major)	5	
15A	Subsystem Ver. Minor)	1	
15C	Win32 Version Value	0	
160	Size of Image	28000	
164	Size of Headers	400	
168	Checksum	0	

The entry point is under .**text** section.

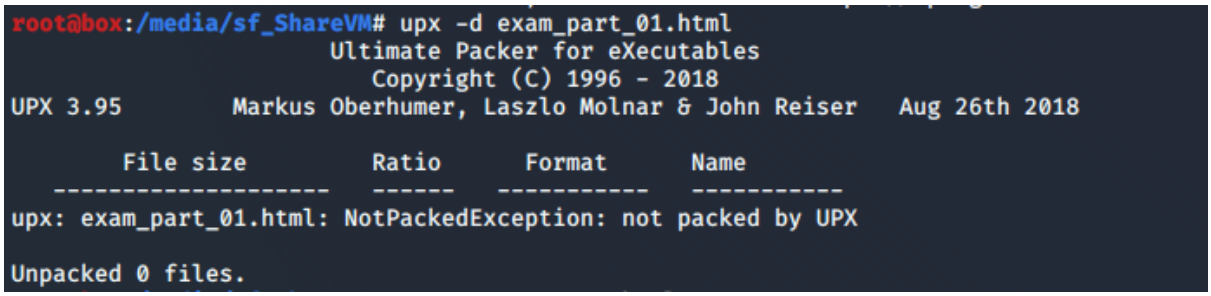
6.What are the sections in the application?



Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size	Characteristics	Ptr to Reloc.	Num. of Reloc.	Num. of Linenum.
.text	400	A800	1000	A617	60000020	0	0	0
.rdata	AC00	5C00	C000	5A0E	40000040	0	0	0
.data	10800	800	12000	12A8	C0000040	0	0	0
.gfids	11000	200	14000	AC	40000040	0	0	0
.rsrc	11200	11400	15000	11268	40000040	0	0	0
.reloc	22600	E00	27000	DE4	42000040	0	0	0

7.Is the entire application packed with UPX?

No, the application is not packed with UPX. If it is , there is .upx0/1 extension.



```
root@box:/media/sf_ShareVM# upx -d exam_part_01.html
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95        Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
upx: exam_part_01.html: NotPackedException: not packed by UPX

Unpacked 0 files.
```

8.Is there any compressed/packed section?

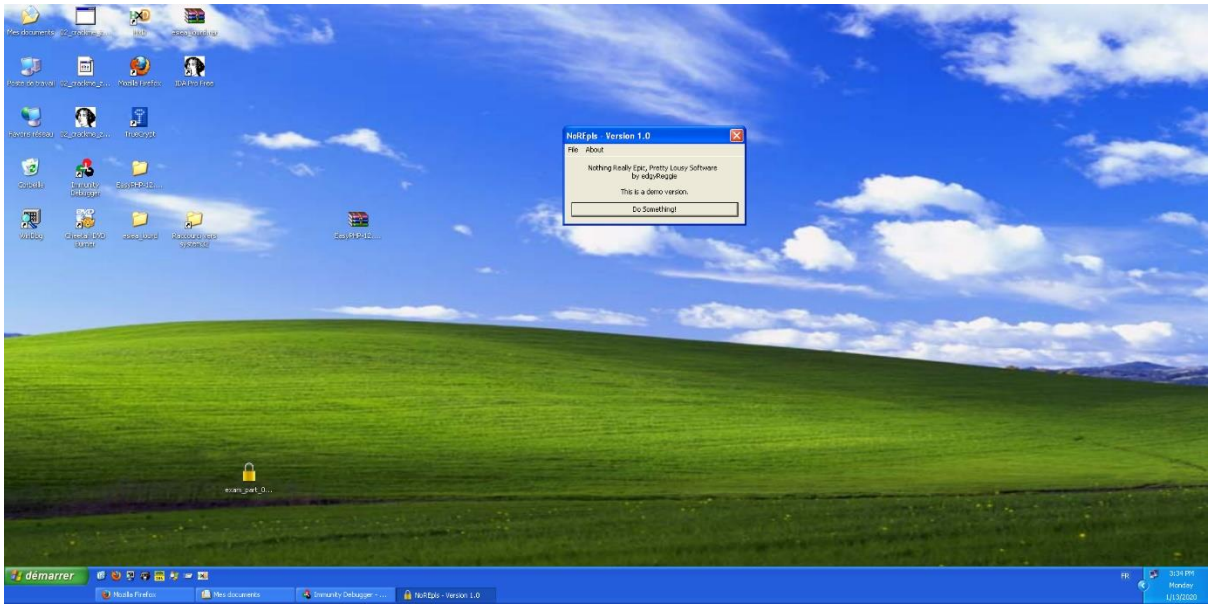
No, there is no compressed / packet section by comparing raw and virtual address of the package.

9.What are the imported libraries? Give 1 API per imported library.

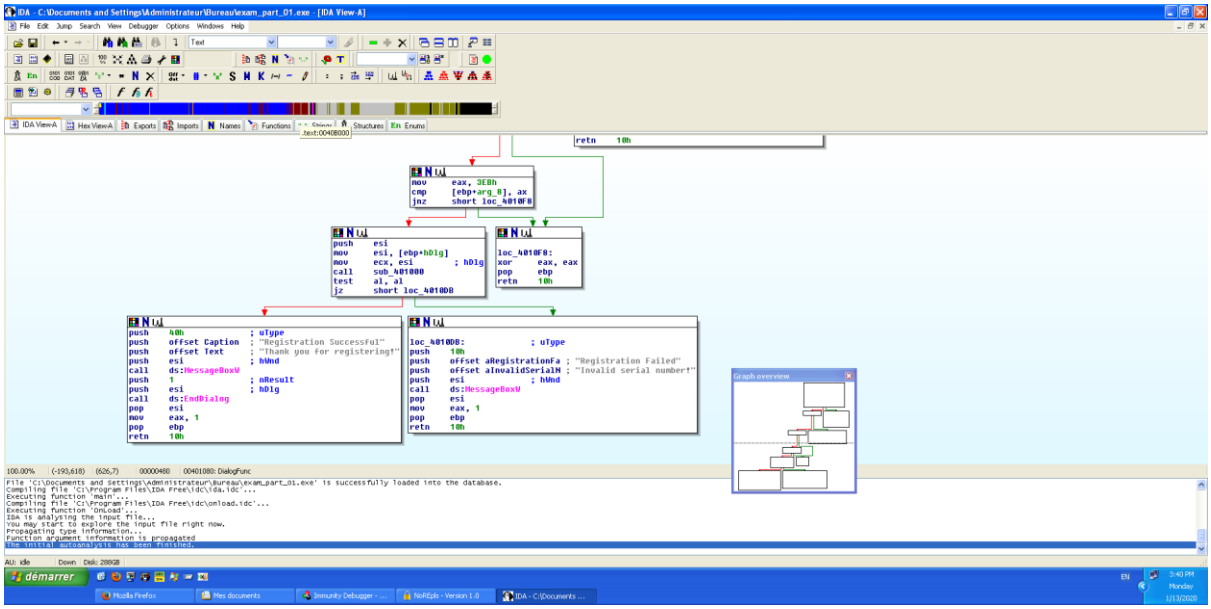
Offset	Name	Func. Count	Bound?	OriginalFirstThunk	TimeDateStamp	Forwarder	NameRVA	FirstThunk
FEC4	USER32.dll	15	FALSE	1141C	0	0	1155E	C108
FED8	COMCTL32.dll	1	FALSE	11314	0	0	1156A	C000
FEEC	KERNEL32.dll	63	FALSE	1131C	0	0	11A00	C008

10. Process of solving by Patching:

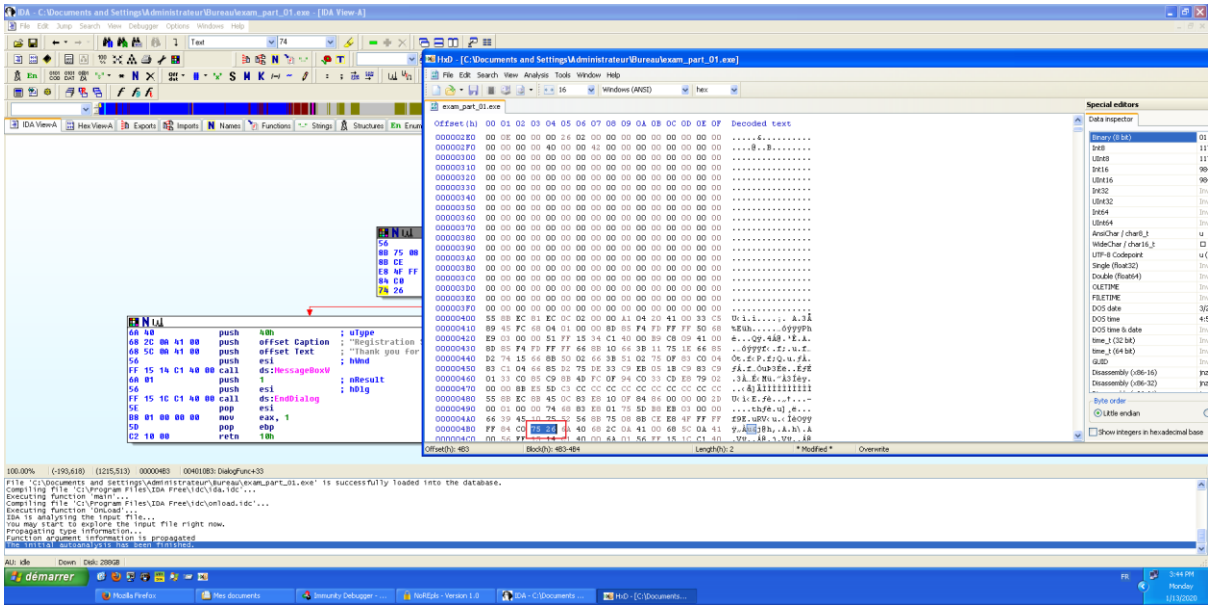
→ We changed the downloaded sample files into .exe file and executed the file to see the overview.



→ We have used IDA tool to patch the software, went to the registration dialogue section.



➔ We edited the opcode, from 74 to 75, so that the jump function will function as our needed way.



→ Also, we found the registration key in strings of the corresponding software.

----Start of Strings Analysis---- (after Unicode covers.)

!This program cannot be run in DOS mode.

Rich/n

.text

`.rdata

@.data

.gfids

@.rsrc

@.reloc

t&j@h,

……

NOREPLS-U89S-N34J-3IOJ-989Y

NoREpls - Version 1.0

Registration Successful

Thank you for registering!

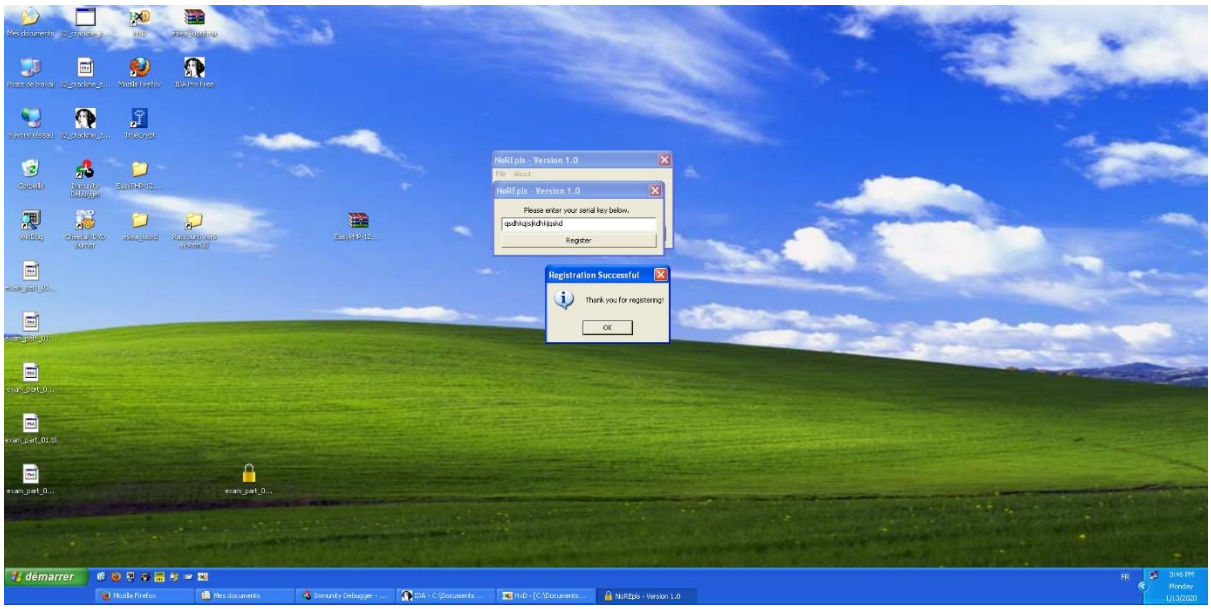
…….

About

----End of Strings Analysis----

Notable Strings: **NOREPLS-U89S-N34J-3IOJ-989Y**

→ After entering the taken strings in dialogue box, the registration is successful.



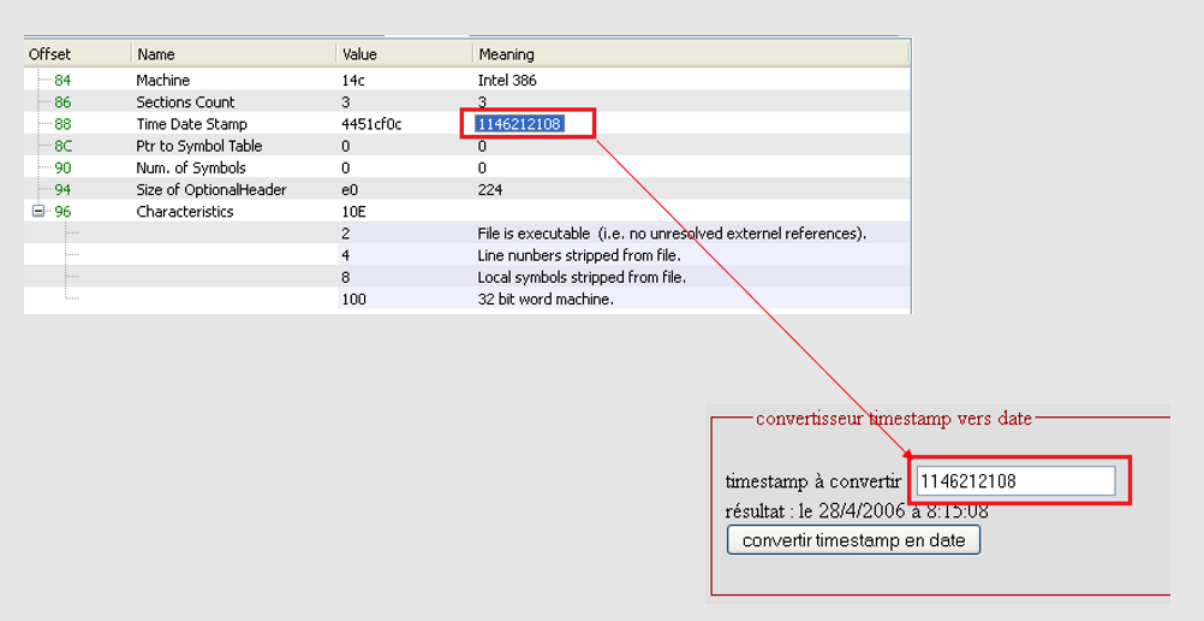
Exam_part_02.html

1.Which compiler was used?

File type : .exe (Executable) – Intel 80386
This was done using linux command through file command and PE Bear.

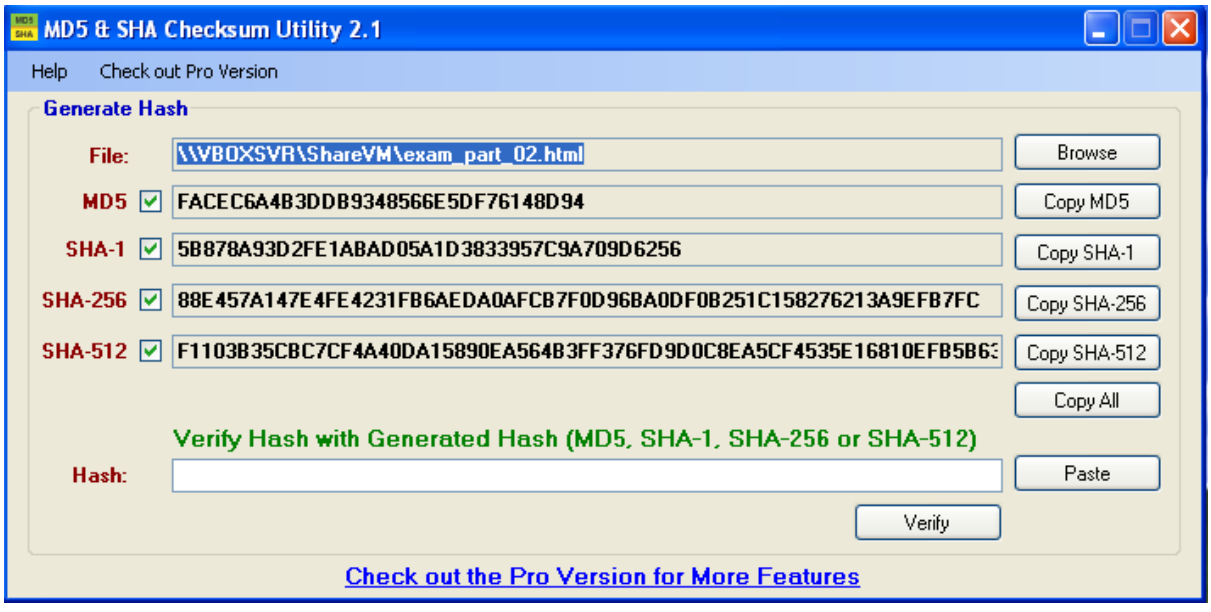
```
root@box:/media/sf_ShareVM# file exam_part_02.html
exam_part_02.html: PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows
```

2.When was this application compiled?



Timestamp: Tuesday, April 28, 2006 08:15:08 AM GMT+02:00 DST
This was followed through under file header options in PE Bear.

3.What is its SHA-1 hash value?

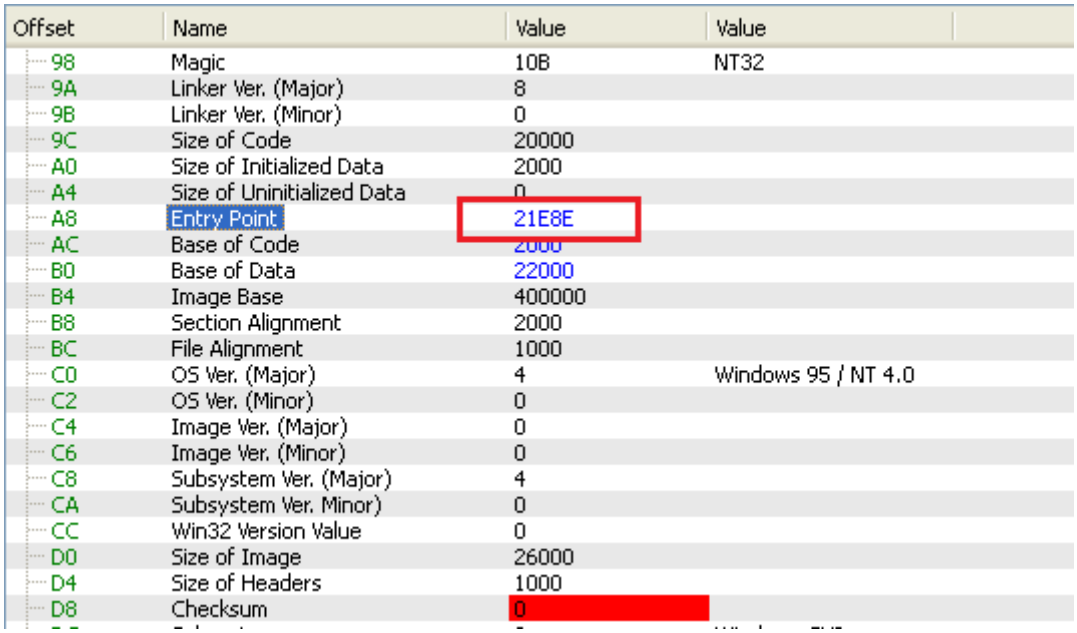


This was done via MD5 and SHA checksum utility

4.Which CPU platform is it compiled for? 32- or 64-bit versions?

Its compiled-on Intel 80386 and It is 32- bit version.

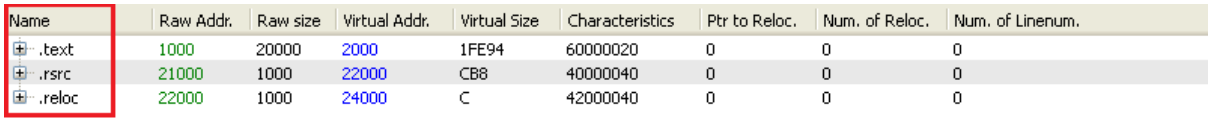
5.What it its entry point? In which section is it?



Offset	Name	Value	Value
98	Magic	10B	NT32
9A	Linker Ver. (Major)	8	
9B	Linker Ver. (Minor)	0	
9C	Size of Code	20000	
A0	Size of Initialized Data	2000	
A4	Size of Uninitialized Data	0	
A8	Entry Point	21E8E	
AC	Base of Code	2000	
B0	Base of Data	22000	
B4	Image Base	400000	
B8	Section Alignment	2000	
BC	File Alignment	1000	
C0	OS Ver. (Major)	4	Windows 95 / NT 4.0
C2	OS Ver. (Minor)	0	
C4	Image Ver. (Major)	0	
C6	Image Ver. (Minor)	0	
C8	Subsystem Ver. (Major)	4	
CA	Subsystem Ver. Minor)	0	
CC	Win32 Version Value	0	
D0	Size of Image	26000	
D4	Size of Headers	1000	
D8	Checksum	0	

The entry point is under .**text** section.

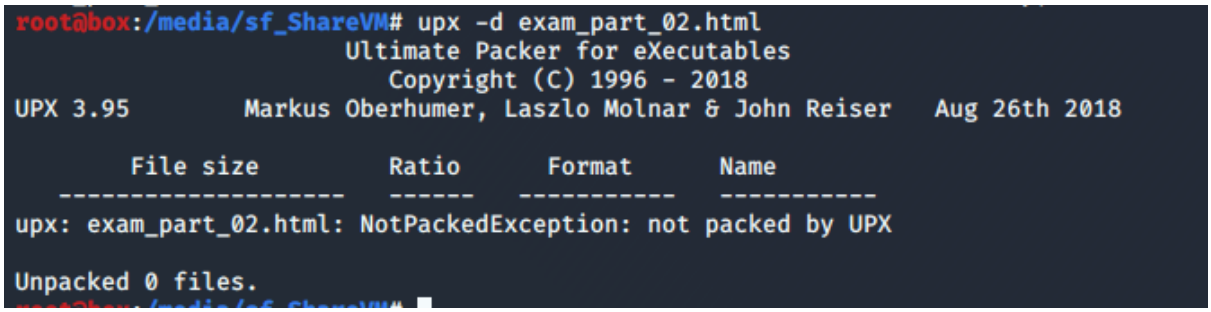
6.What are the sections in the application?



Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size	Characteristics	Ptr to Reloc.	Num. of Reloc.	Num. of Linenum.
.text	1000	20000	2000	1FE94	60000020	0	0	0
.rsrc	21000	1000	22000	CB8	40000040	0	0	0
.reloc	22000	1000	24000	C	42000040	0	0	0

7.Is the entire application packed with UPX?

No, the application is not packed with UPX. If it is , there is .upx0/1 extension.



```
root@box:/media/sf_ShareVM# upx -d exam_part_02.html
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95        Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
upx: exam_part_02.html: NotPackedException: not packed by UPX

Unpacked 0 files.
```

8.Is there any compressed/packed section?

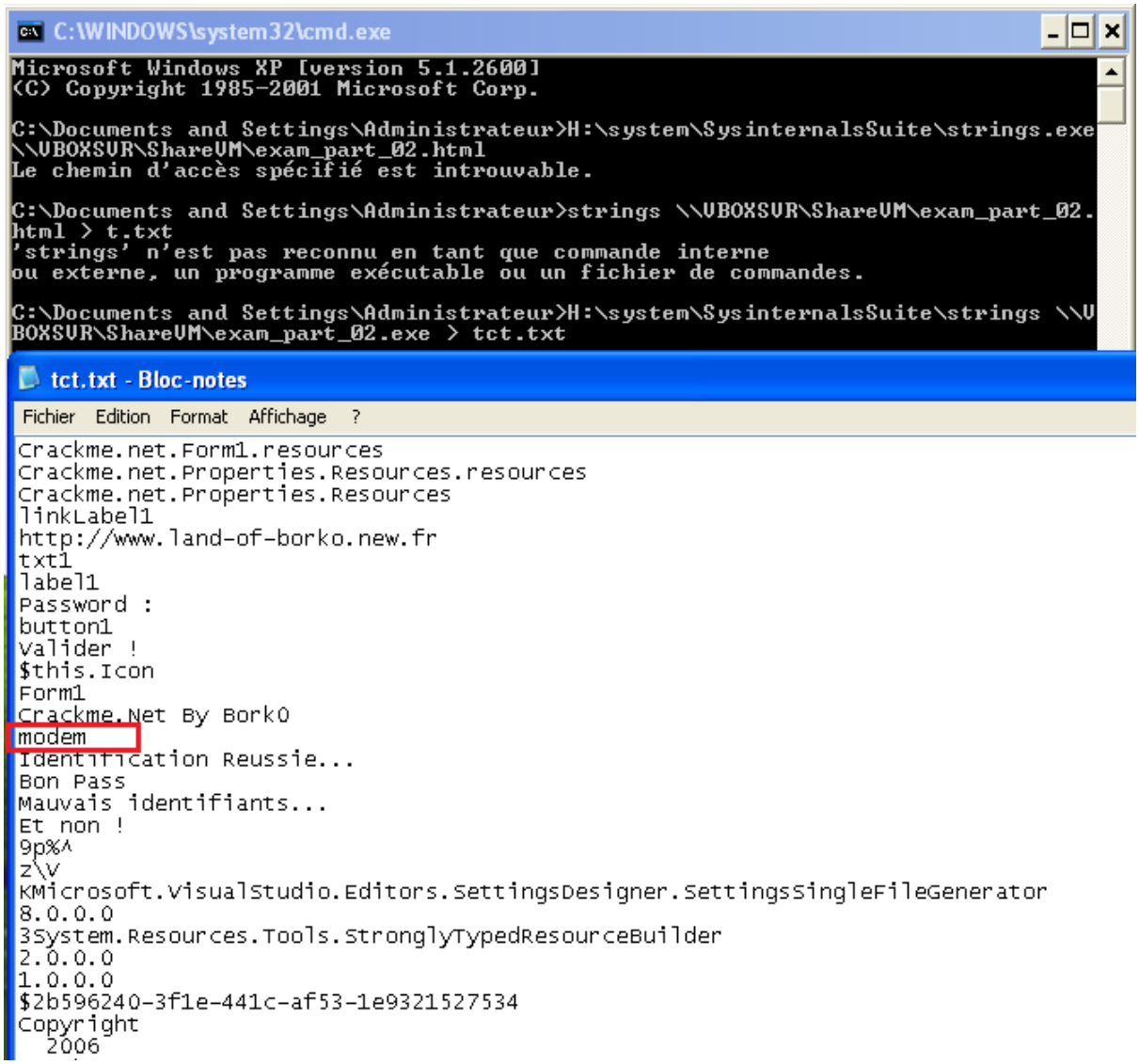
No, there is no compressed / packet section by comparing raw and virtual address of the package.

9.What are the imported libraries? Give 1 API per imported library.

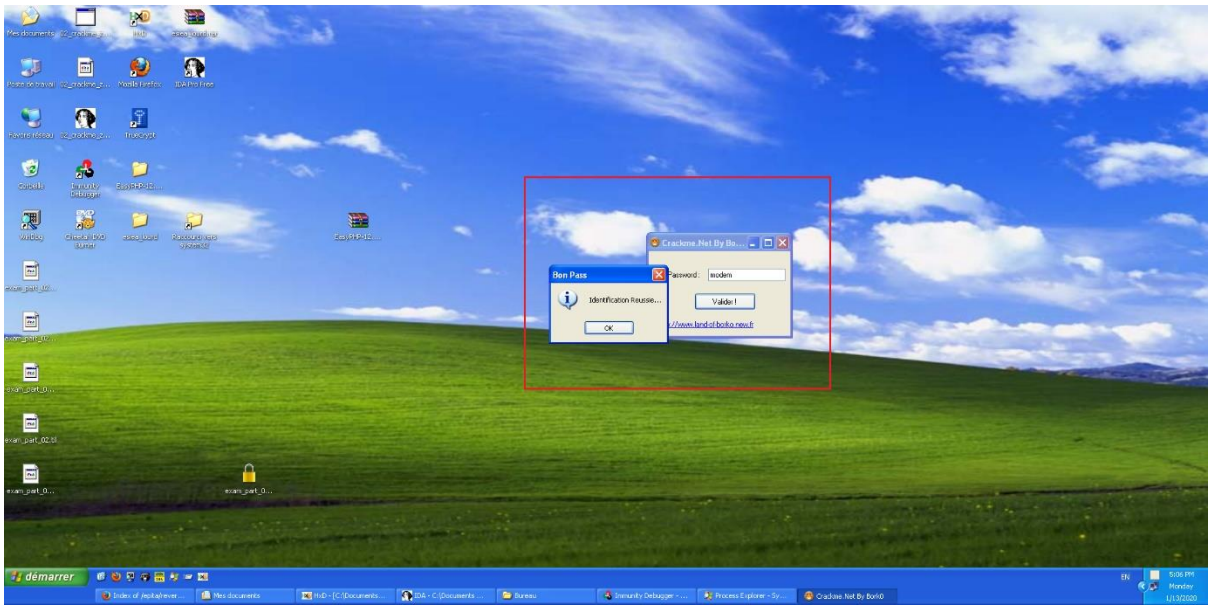
Offset	Name	Func. Count	Bound?	OriginalFirstThunk	TimeDateStamp	Forwarder	NameRVA	FirstThunk
20E3C	mscoree.dll	1	FALSE	21E64	0	0	21E7E	2000

10. Process of solving by Patching:

→ By analysing the strings before the identification reussie, found the password as Modem.



→ After entering the taken strings in dialogue box, the identification is successful.



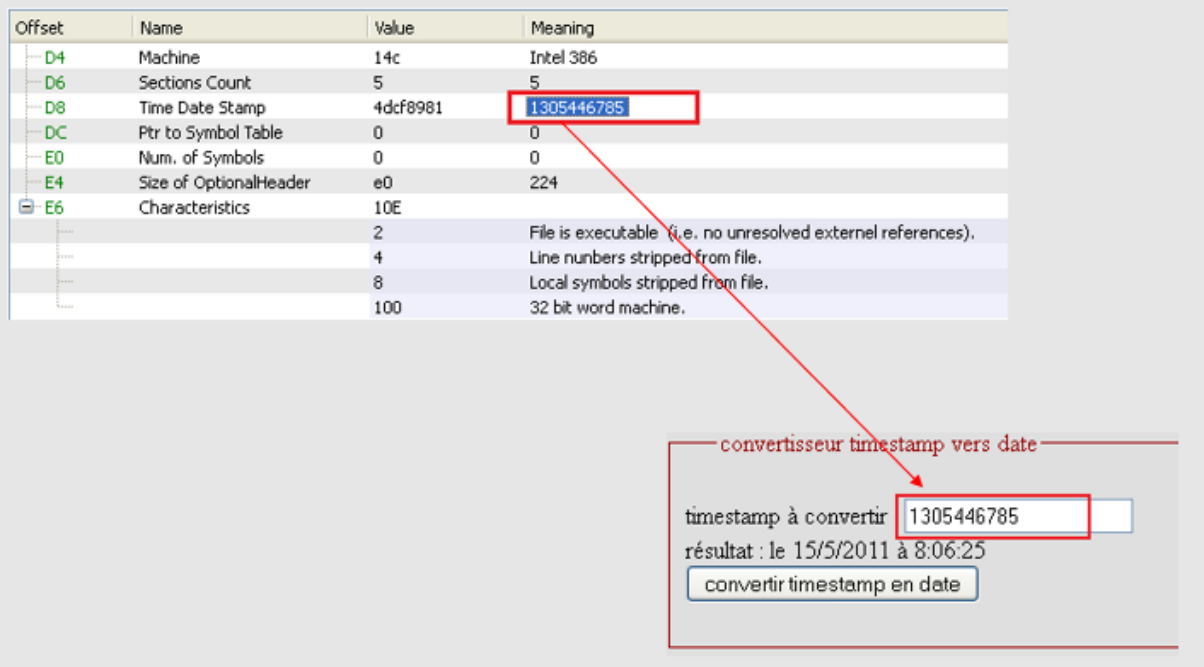
Exam_part_03.html

1.Which compiler was used?

File type : .exe (Executable) – Intel 80386
This was done using linux command through file command and PE Bear.

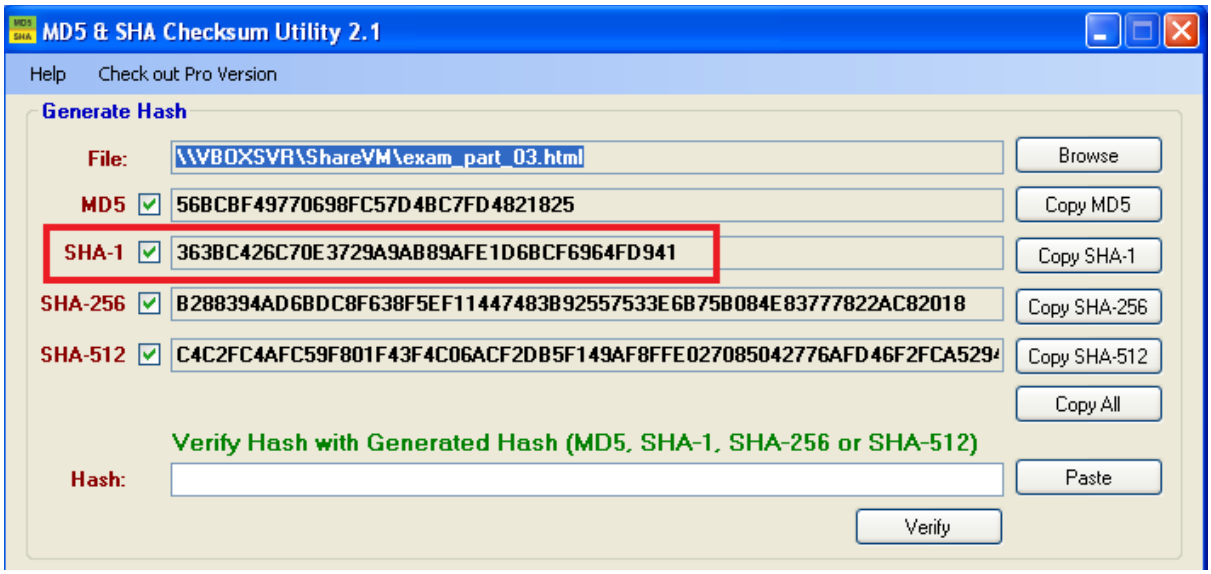
```
root@box:/media/sf_ShareVM# file exam_part_03.html
exam_part_03.html: PE32 executable (console) Intel 80386, for MS Windows
```

2.When was this application compiled?



Timestamp: Tuesday, May 05, 2011 08:06:25 AM GMT+02:00 DST
This was followed through under file header options in PE Bear.

3.What is its SHA-1 hash value?



This was done via MD5 and SHA checksum utility

4.Which CPU platform is it compiled for? 32- or 64-bit versions?

Its compiled-on Intel 80386 and It is 32- bit version.

5.What it its entry point? In which section is it?

Offset	Name	Value	Value
E8	Magic	10B	NT32
EA	Linker Ver. (Major)	6	
EB	Linker Ver. (Minor)	0	
EC	Size of Code	26000	
F0	Size of Initialized Data	9000	
F4	Size of Uninitialized Data	0	
F8	Entry Point	4700	
FC	Base of Code	1000	
100	Base of Data	1000	
104	Image Base	400000	
108	Section Alignment	1000	
10C	File Alignment	1000	
110	OS Ver. (Major)	4	Windows 95 / NT 4.0
112	OS Ver. (Minor)	0	

The entry point is under .**text** section.

6.What are the sections in the application?

Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size	Characteristics	Ptr to Reloc.	Num. of Reloc.	Num. of Linenum.
.text	1000	26000	1000	25950	60000020	0	0	0
.rdata	27000	2000	27000	1CF0	40000040	0	0	0
.data	29000	2000	29000	34F0	C0000040	0	0	0
.idata	2B000	1000	2D000	757	C0000040	0	0	0
.reloc	2C000	2000	2E000	10B1	42000040	0	0	0

7.Is the entire application packed with UPX?

No, the application is not packed with UPX. If it is, there is .upx0/1 extension.

```
root@box:/media/sf_ShareVM# upx -d exam_part_03.html
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95        Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
upx: exam_part_03.html: NotPackedException: not packed by UPX

Unpacked 0 files.
```

8.Is there any compressed/packed section?

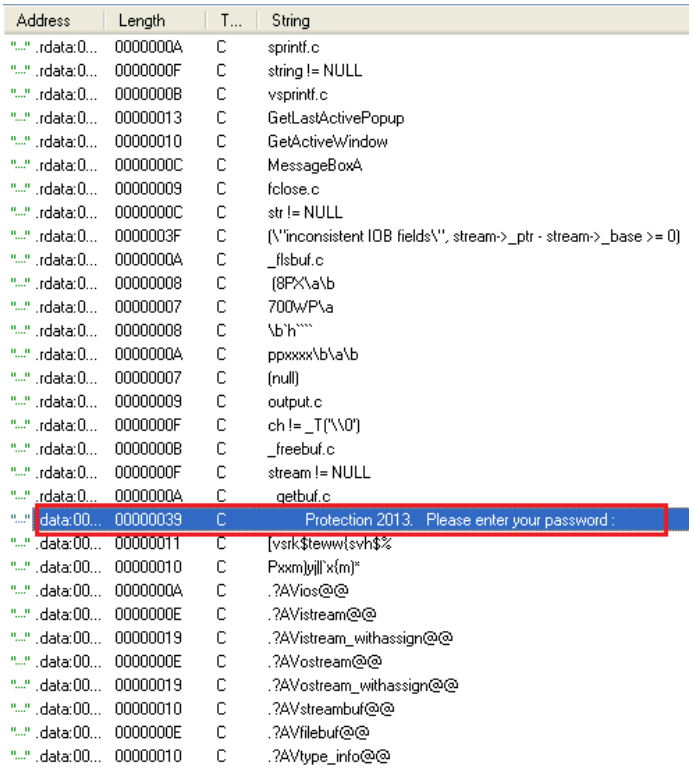
No, there is no compressed / packet section by comparing raw and virtual address of the package.

9.What are the imported libraries? Give 1 API per imported library.

Offset	Name	Func. Count	Bound?	OriginalFirstThunk	TimeDateStamp	Forwarder	NameRVA	FirstThunk
2B000	KERNEL32.dll	51	FALSE	2D028	0	0	2D5EE	2D148

10. Process of solving by Patching:

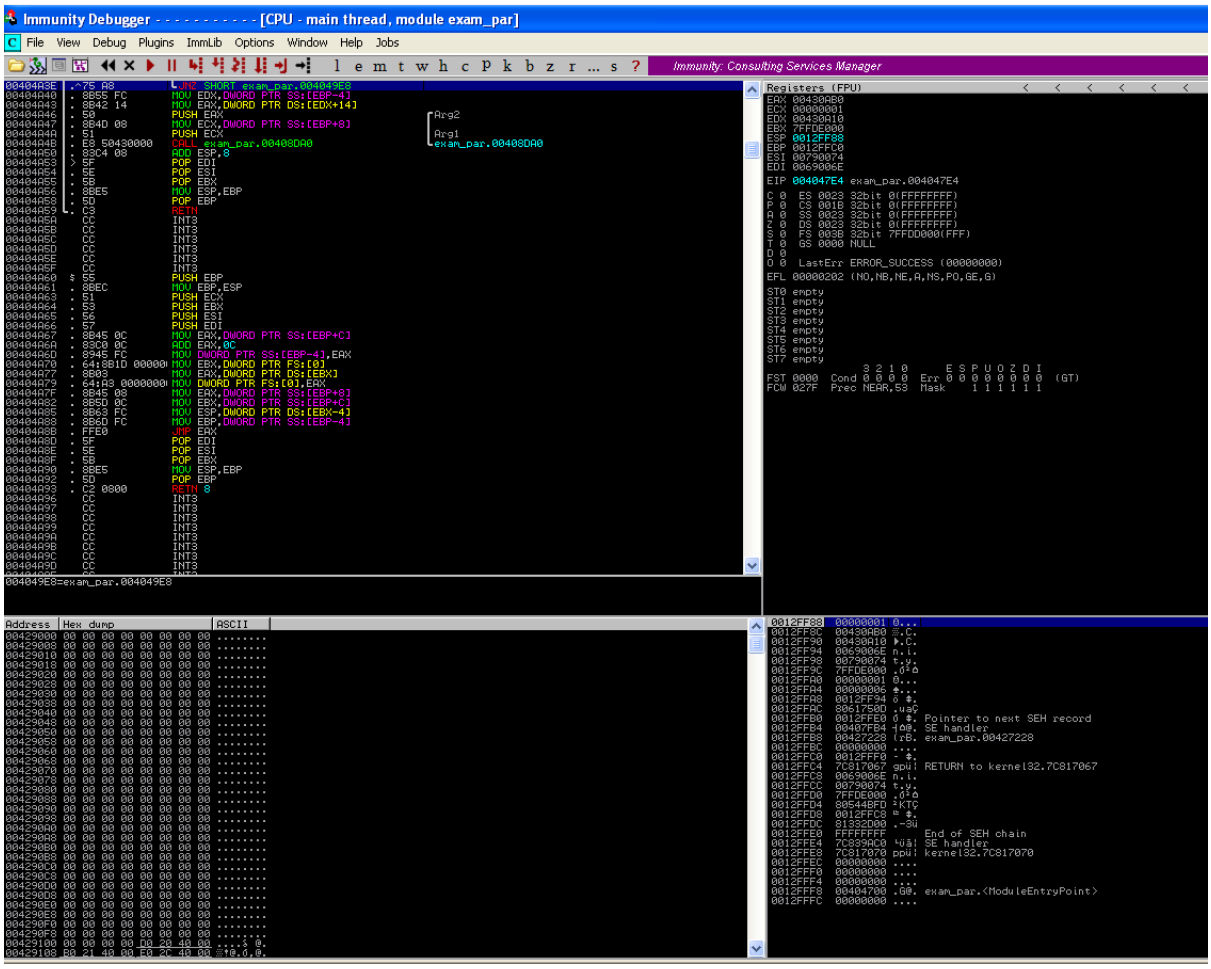
The Exam_part_03.html file is opened using IDA Pro to view the structure of the complied assembly program in the string sections which offers the flow of the executable file.



Address	Length	T...	String
".." .rdata:0...	0000000A	C	sprintf.c
".." .rdata:0...	0000000F	C	string != NULL
".." .rdata:0...	0000000B	C	vsprintf.c
".." .rdata:0...	00000013	C	GetLastActivePopup
".." .rdata:0...	00000010	C	GetActiveWindow
".." .rdata:0...	0000000C	C	MessageBoxA
".." .rdata:0...	00000009	C	fclose.c
".." .rdata:0...	0000000C	C	str != NULL
".." .rdata:0...	0000003F	C	(\"inconsistent IOB fields\", stream->_ptr - stream->_base >= 0)
".." .rdata:0...	0000000A	C	_flsbuf.c
".." .rdata:0...	00000008	C	(8PX\a\b
".." .rdata:0...	00000007	C	700WP\a
".." .rdata:0...	00000008	C	\b`h````
".." .rdata:0...	0000000A	C	ppxxxx\b\a\b
".." .rdata:0...	00000007	C	(null)
".." .rdata:0...	00000009	C	output.c
".." .rdata:0...	0000000F	C	ch != _T('\\0')
".." .rdata:0...	0000000B	C	_freebuf.c
".." .rdata:0...	0000000F	C	stream != NULL
".." .rdata:0...	0000000A	C	getbuf.c
".." data:00...	00000039	C	Protection 2013. Please enter your password :
".." .data:00...	00000011	C	[vsrk$teww{svh$%
".." .data:00...	00000010	C	Pxxm}yjl`x{m}*
".." .data:00...	0000000A	C	.?AVios@@
".." .data:00...	0000000E	C	.?AVistream@@
".." .data:00...	00000019	C	.?AVistream_withassign@@
".." .data:00...	0000000E	C	.?AVostream@@
".." .data:00...	00000019	C	.?AVostream_withassign@@
".." .data:00...	00000010	C	.?AVstreambuf@@
".." .data:00...	0000000E	C	.?AVfilebuf@@
".." .data:00...	00000010	C	.?AVtype_info@@

Where we can get the virtual address of the password given location.

The x64 debugger is opened and initially we run the file which executes by offsets F8 key:



By examining the hex of the keys in the memory block, it is in decrypted form