

정보보안 실무과제

담당 교수 : 이병천 교수님

91812488 성우상
91812866 이승훈
91812139 김성준

Contents

1. Web hacking

2. Reversing

*****정보보안과제*****

JB UNIVERSITY

webhacking & reversing

member

HACKER

VIRUS LOADING....

*****정보보안과제*****

JB UNIVERSITY

web & reversing

menu member

91812139김성준-Reversing

91812866이승훈-Webhacking

91812488성우상-Reversing

이 페이지 내용:

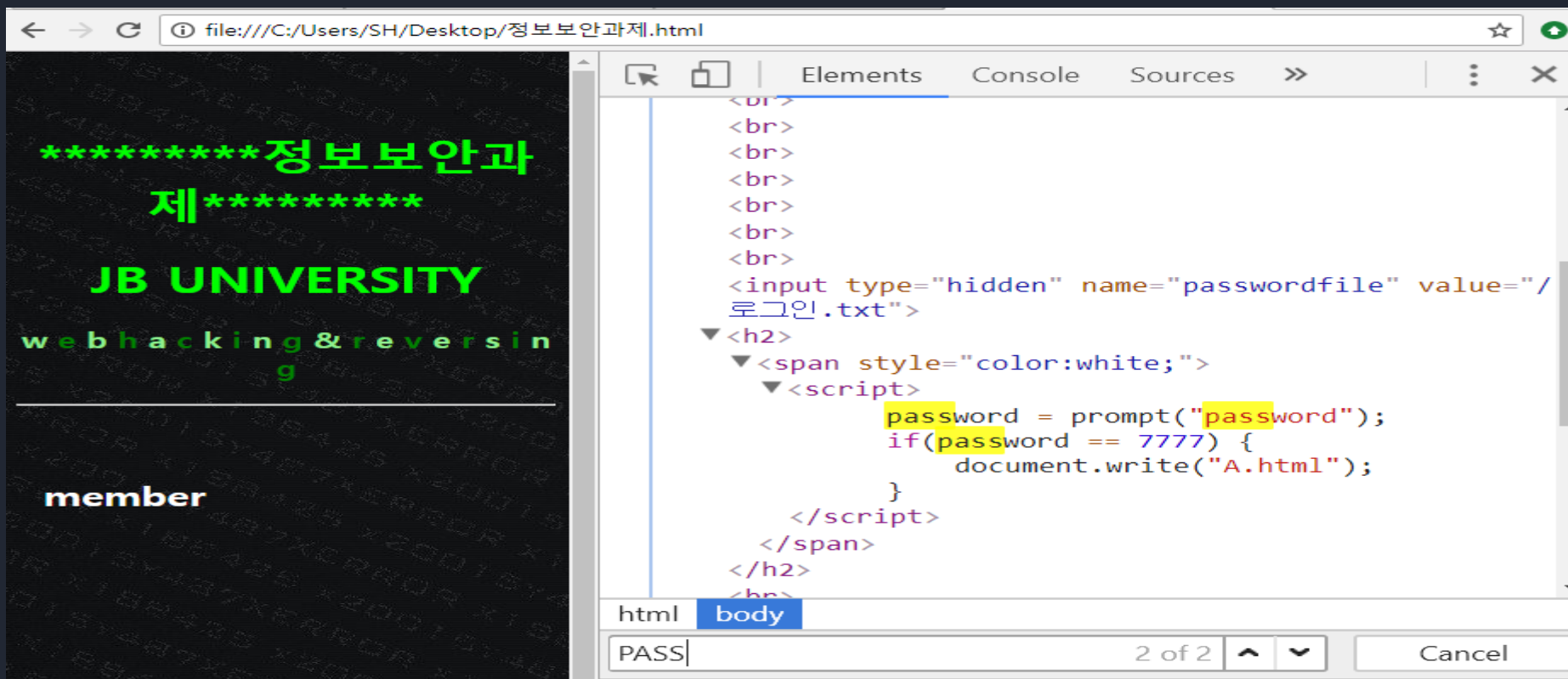
password

확인

취소

페이지를 들어가보면 이런 문구가 뜬다. 보아하니 비밀번호를 찾아야하는 것 같다.

Web & Reversing



F12를 누르면 개발자 도구가 뜬다. 여기서 ctrl F를 누른다음 pass단어를 찾아 보자. 찾아보니 비밀번호가 나와있다.



입력을 해보니 가운데에 A.html이라는 문구가 나타났다. 주소창에 쳐서 들어가보자.

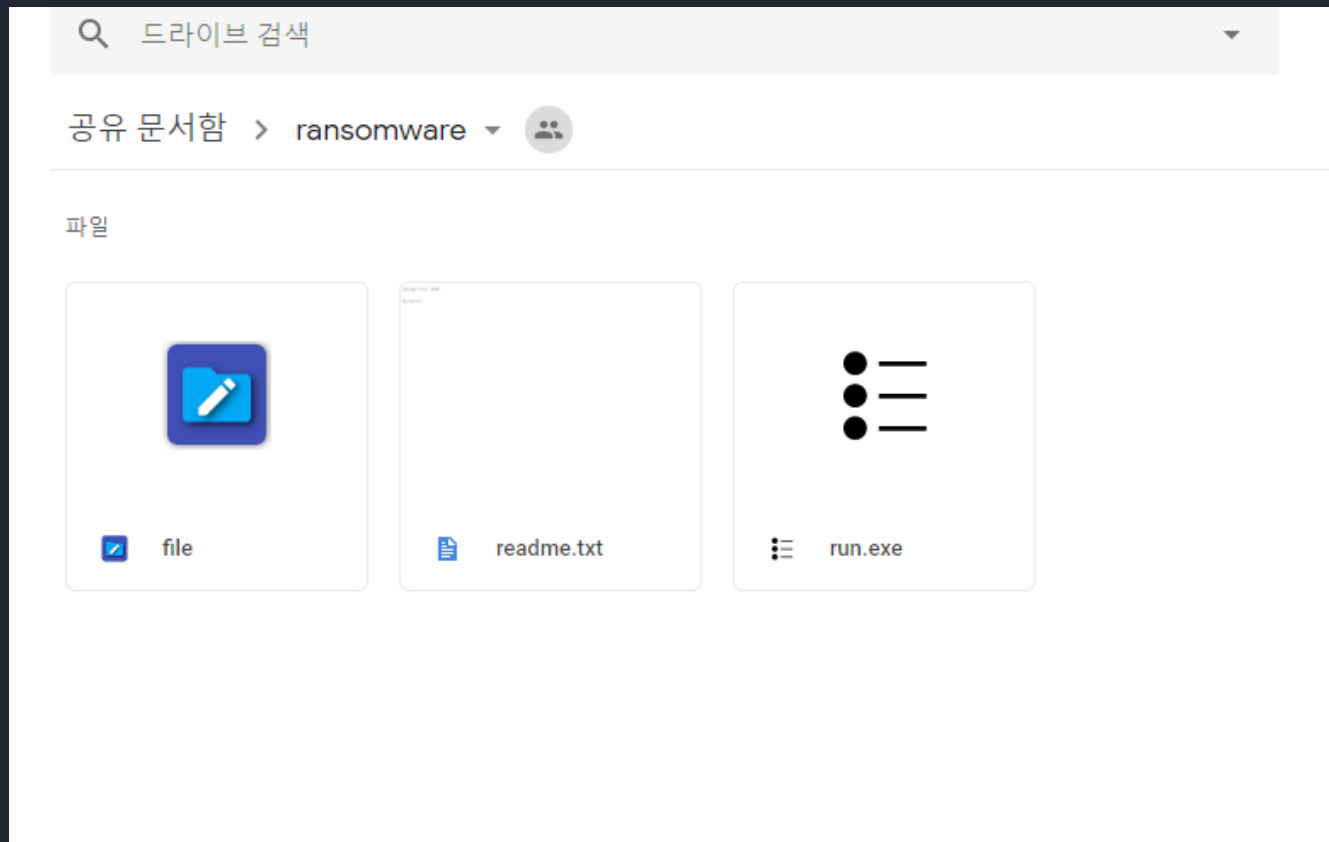
Web & Reversing



새로운 창이 열렸다.



Click을 눌러보면 창이 나타날 것이다.






다운받는다.

리버싱(Rreverse Code Engineering)이란?

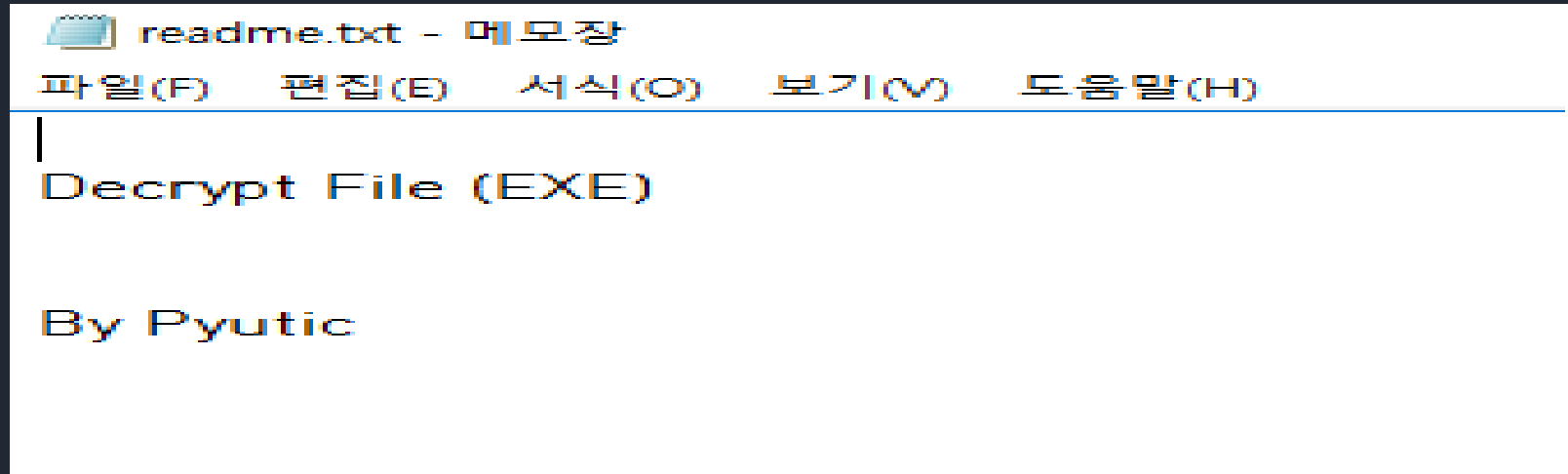
- 리버스(reverse-뒤집다)와 엔지니어링(engineering-공학기술)의 합성어로 리버싱이라고 줄여 말한다.

#리버싱 기술

- 완성된 프로그램을 거꾸로 분석하여 설계도를 추출
- 완성된 시스템을 역추적 (역공학)
- 소프트웨어의 유지보수
- 프로그램 동작변경
- 복제프로그램 개발(대표적인 크랙)
- 향상된 프로그램 개발
- 악성 코드 분석(보안)

 file	2012-03-04 오후...	파일	9KB
 readme.txt	2012-03-12 오전...	텍스트 문서	1KB
 run.exe	2012-03-04 오후...	응용 프로그램	10KB

주어진 zip파일을 풀면 이렇게 세 개의 파일이 나온다.



일단 readme.txt 부터 열어보면 파일을 Decrypt 하란다. 파일 종류는 exe인 것 같다.

C:\Users\성준\Desktop\리버싱기초 이해\ransomware\run.exe

```
나는 나쁜놈이다!  
나는 매우 나쁘기 때문에 너의 파일을 암호화했다!  
너의 파일을 복구하고 싶다면 5천억 달러를 입금하고 받은 키값으로 파일을 복구해라!  
  
<ey :
```

run.exe 을 실행하면 이런 화면이 뜬다. Key를 입력하면 그 Key를 가지고 Decrypt 하는 건가 보다. 5천억을 진짜 입금할 순 없으니 살펴보자.

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	DE	C0	1B	8C	8C	93	9E	86	98	97	9A	8C	73	6C	9A	8B	PÀ.GE"Žt~—šGEslš<
00000010	34	8F	93	9E	86	9C	97	9A	CC	8C	93	9A	8B	8C	8F	93	4."Žt+œ—šİGE"š<E."
00000020	9E	86	9C	97	9A	8C	8C	93	9A	8B	8C	8F	93	9E	86	9C	Žt+œ—šGE"š<E."Žt+œ
00000030	97	9A	8C	8C	93	9A	8B	8C	8F	93	9E	86	6C	97	9A	8C	—šGE"š<E."Žt+œ—šGE
00000040	82	8C	20	85	8C	3B	9A	53	A7	24	96	D6	41	AD	C7	F2	,E...E;šSS\$—ÖA.Çð
00000050	E2	FF	AF	E3	EC	E9	FB	E5	FB	E1	AC	F0	FB	E5	E2	E0	äÿ—äiëüäüä—öüäâä
00000060	E7	BE	E4	F9	B7	E8	F9	E2	B3	F3	E5	AC	CB	DC	CD	A6	çäü—èùä—óä—EÜí;
00000070	F1	F8	FE	E9	A2	9E	97	81	A8	8F	93	9E	86	9C	97	9A	fløpécž—. ."Žt+œ—š
00000080	81	CD	37	0A	C2	AC	45	50	D7	A6	56	54	D3	AC	46	50	.í7.Ä—EP×;VTÓ—FP
00000090	40	E5	DE	4C	DB	BE	4C	5F	B1	CC	DA	4F	D9	BA	41	4F	@âpLÜ×L±İÚOÜ°AO
000000A0	A9	C5	FE	45	C7	B7	50	4F	CC	CB	C3	48	C7	AF	59	5D	@ÂpEÇ—PÖİĖĀHÇ—Y]
000000B0	CF	BC	5C	59	FD	AC	59	59	AD	DA	EE	50	DE	A6	56	54	İ4\YŸ—YY.ÜİPß;VT
000000C0	BC	DA	DC	50	D2	AB	46	4C	B5	C8	D1	5F	DF	BA	46	4F	¼ÚÜPÖ«FLpĖĒ°FO
000000D0	C1	F3	E8	E4	C6	B3	54	45	9C	97	9A	8C	8C	93	9A	8B	ÁóèäE°TEœ—šGE"š<
000000E0	8C	8F	93	9E	86	9C	97	9A	8C	8C	93	9A	8B	8C	8F	93	E."Žt+œ—šGE"š<E."
000000F0	CE	C3	9C	97	D6	8D	8F	93	26	A0	DF	C0	93	9E	86	9C	İÄœ—Ö... "œ BÄ"Žt+œ
00000100	97	9A	8C	8C	73	9A	88	8D	84	92	94	86	9C	B7	9A	8C	—šGEsš^...' "tœ—šGE
00000110	8C	83	9A	8B	8C	2F	92	9E	36	55	96	9A	8C	3C	92	9A	Efš<E/'ž6U—šE<'š
00000120	8B	5C	8E	93	9E	86	DC	97	9A	9C	8C	93	9A	89	8C	8F	<\ž"Žt+Ü—šœE"šŋE.
00000130	96	9E	87	9C	97	9A	8C	8C	96	9A	8A	8C	8F	93	9E	86	—žt+œ—šGE—šŠE."Žt
00000140	9C	77	9B	8C	8C	83	9A	8B	8C	8F	93	9E	85	9C	D7	1B	œw>GEfš<E."ž...œ×.
00000150	8C	8C	83	9A	8B	9C	8F	93	9E	86	8C	97	9A	9C	8C	93	GEfš<E."Žt+œ—šœE"
00000160	9A	8B	8C	8F	83	9E	86	9C	97	9A	8C	8C	93	9A	8B	8C	š<E.f.žt+œ—šGE—šœE"
00000170	7B	42	9F	86	40	97	9A	8C	8C	43	9B	8B	78	8E	93	9E	{BŸt@—šGEc>xž"ž
00000180	86	9C	97	9A	8C	8C	93	9A	8B	8C	8F	93	9E	86	9C	97	t+œ—šGE"š<E."Žt+œ—
00000190	9A	8C	8C	93	9A	8B	8C	8F	93	9E	86	9C	97	9A	8C	8C	šGE"š<E."Žt+œ—šGE
000001A0	93	9A	8B	8C	8F	93	9E	86	9C	97	9A	8C	8C	93	9A	8B	"š<E."Žt+œ—šGE"š<
000001B0	8C	8F	93	9E	86	9C	97	9A	8C	8C	93	9A	8B	8C	8F	93	E."Žt+œ—šGE"š<E."
000001C0	9E	86	9C	97	9A	8C	8C	93	9A	8B	8C	8F	93	9E	86	9C	žt+œ—šGE"š<E."Žt+œ
000001D0	97	9A	8C	8C	93	9A	8B	8C	8F	93	9E	86	9C	97	9A	8C	—šGE"š<E."Žt+œ—šGE
000001E0	8C	93	9A	8B	8C	8F	93	9E	D3	CC	CF	AA	8C	8C	93	9A	E"š<E."žÓİ—šGE"š
000001F0	8B	2C	8E	93	9E	96	9C	97	9A	8C	8C	93	9A	8F	8C	8F	<,ž"ž—œ—šGE"š.E.

널 패딩이 있을 만한곳에 0xD만큼 특정 값이 반복되는 것을 볼 수 있었다. 따라서 키값은 0xD이다.

Web & Reversing

[- CPU - main thread, module run]

File View Debug Plugins Options Window Help

⏮ ⏪ ⏩ ⏭ 🔍 ?

Address	Hex dump	Disassembly	Comment
00ECDE	00	DB 00	
00ECDF	0060BE00	DD run.00BE6000	
00ECE3	D0	DB D0	
00ECE4	EC	DB EC	
00ECE5	0080BE00	DD run.00BE8000	
00ECE9	40	DB 40	CHAR 'e'
00ECEA	53	DB 53	CHAR 'S'
00ECEB	FF	DB FF	
00ECEC	57	PUSH EDI	
00ECED	EB 0B	JMP SHORT run.00ECCFA	
00ECF0	90	NOP	
00ECF1	8A06	MOV AL, BYTE PTR DS:[ESI]	
00ECF2	46	INC ESI	
00ECF3	8807	MOV BYTE PTR DS:[EDI], AL	
00ECF5	47	INC EDI	
00ECF6	01DB	ADD EBX, EBX	
00ECF8	75 07	JNZ SHORT run.00ECD01	
00ECFA	8B1E	MOV EBX, DWORD PTR DS:[ESI]	
00ECFC	83EE FC	SUB ESI, -4	
00ECFF	11DB	ADC EBX, EBX	
00ECD01	72 ED	JB SHORT run.00ECCF0	
00ECD03	B8 01000000	MOV EAX, 1	
00ECD08	01DB	ADD EBX, EBX	
00ECD0A	75 07	JNZ SHORT run.00ECD13	
00ECD0C	8B1E	MOV EBX, DWORD PTR DS:[ESI]	
00ECD0E	83EE FC	SUB ESI, -4	
00ECD11	11DB	ADC EBX, EBX	
00ECD13	11C0	ADC EAX, EAX	
00ECD15	01DB	ADD EBX, EBX	
00ECD17	73 0B	JNB SHORT run.00ECD24	
00ECD19	75 28	JNZ SHORT run.00ECD43	
00ECD1B	8B1E	MOV EBX, DWORD PTR DS:[ESI]	
00ECD1D	83EE FC	SUB ESI, -4	
00ECD20	11DB	ADC EBX, EBX	
00ECD22	72 1F	JB SHORT run.00ECD43	
00ECD24	48	DEC EAX	
00ECD25	01DB	ADD EBX, EBX	
00ECD27	75 07	JNZ SHORT run.00ECD30	
00ECD29	8B1E	MOV EBX, DWORD PTR DS:[ESI]	
00ECD2B	83EE FC	SUB ESI, -4	

Registers (FPU)

EAX 68242EEA
ECX 00ECECE0 run.<ModuleEntr
EDX 00ECECE0 run.<ModuleEntr
EBX 00258000
ESP 0019FF84
EBP 0019FF94
ESI 00ECECE0 run.<ModuleEntr
EDI 00ECECE0 run.<ModuleEntr
EIP 00ECECE0 run.<ModuleEntr

C 0 ES 002B 32bit 0(FFFFFFF)
P 1 CS 0023 32bit 0(FFFFFFF)
A 0 SS 002B 32bit 0(FFFFFFF)
Z 1 DS 002B 32bit 0(FFFFFFF)
S 0 FS 0053 32bit 25B000(F)
T 0 GS 002B 32bit 0(FFFFFFF)
D 0
0 0 LastErr ERROR_SUCCESS
EFL 0000246 (NO,NB,E,BE,NS

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0

FST 0000 Cond 0 0 0 0 Err
FCW 027F Prec NEAR,53 Mask

Address	Hex dump	Disassembly	Comment
00ECF000	0000	ADD BYTE PTR DS:[EAX], AL	
00ECF002	0000	ADD BYTE PTR DS:[EAX], AL	
00ECF004	0000	ADD BYTE PTR DS:[EAX], AL	
00ECF006	0000	ADD BYTE PTR DS:[EAX], AL	
00ECF008	04 00	ADD AL, 0	
00ECF00A	0000	ADD BYTE PTR DS:[EAX], AL	
00ECF00C	0000	ADD BYTE PTR DS:[EAX], AL	

0019FF84 76D38484 RETURN to KERNEL
0019FF88 00258000
0019FF8C 76D38460 KERNEL32.BaseThr
0019FF90 68242EEA
0019FF94 0019FFDC
0019FF98 77AC2EC0 RETURN to ntdll.
0019FF9C 00258000
0019FFA0 3B614214

올리디버거로 분석해보니 entry point가 pushad 로 시작한다 pushad로 시작하니깐 아마 upx로 패킹 했을거라 생각한다.

Address	Hex dump	Disassembly	Comment
00ECF000	0000	ADD BYTE PTR DS:[EAX],AL	
00ECF002	0000	ADD BYTE PTR DS:[EAX],AL	
00ECF004	0000	ADD BYTE PTR DS:[EAX],AL	
00ECF006	0000	ADD BYTE PTR DS:[EAX],AL	
00ECF008	04 00	ADD AL,0	
00ECF00A	0000	ADD BYTE PTR DS:[EAX],AL	
00ECF00C	0000	ADD BYTE PTR DS:[EAX],AL	
00ECF00E	0100	ADD DWORD PTR DS:[EAX],EAX	
00ECF010	1800	SBB BYTE PTR DS:[EAX],AL	
00ECF012	0000	ADD BYTE PTR DS:[EAX],AL	
00ECF014	1800	SBB BYTE PTR DS:[EAX],AL	

Web & Reversing

File View Debug Plugins Options Window Help

LEMTW H C / K B R ... S ?

0044A774	5B	POP EBX	
0044A775	68 ACC14400	PUSH run.0044C1AC	ASCII "Key : "
0044A77A	FF15 B0C04400	CALL DWORD PTR DS:[44C0B01]	MSVCR100.printf
0044A780	83C4 04	ADD ESP,4	
0044A783	E8 7868FBFF	CALL run.00401000	
0044A788	68 70D34400	PUSH run.0044D370	
0044A78D	68 B4C14400	PUSH run.0044C1B4	ASCII "%s"
0044A792	FF15 B8C04400	CALL DWORD PTR DS:[44C0B81]	MSVCR100.scanf
0044A798	83C4 08	ADD ESP,8	
0044A79B	C745 E8 70D34400	MOV DWORD PTR SS:[EBP-18],run.0044D370	
0044A7A2	8B45 E8	MOV EAX,DWORD PTR SS:[EBP-18]	
0044A7A5	83C0 01	ADD EAX,1	
0044A7A8	8945 E4	MOV DWORD PTR SS:[EBP-1C],EAX	
0044A7AB	8B4D E8	MOV ECX,DWORD PTR SS:[EBP-18]	
0044A7AE	8A11	MOV DL,BYTE PTR DS:[ECX]	
0044A7B0	8855 E3	MOV BYTE PTR SS:[EBP-1D],DL	
0044A7B3	8345 E8 01	ADD DWORD PTR SS:[EBP-18],1	
0044A7B7	807D E3 00	CMP BYTE PTR SS:[EBP-1D],0	
0044A7BB	75 EE	JNZ SHORT run.0044A7AB	
0044A7BD	8B45 E8	MOV EAX,DWORD PTR SS:[EBP-18]	
0044A7C0	2B45 E4	SUB EAX,DWORD PTR SS:[EBP-1C]	
0044A7C3	8945 DC	MOV DWORD PTR SS:[EBP-24],EAX	
0044A7C6	8B4D DC	MOV ECX,DWORD PTR SS:[EBP-24]	
0044A7C9	894D F4	MOV DWORD PTR SS:[EBP-C],ECX	
0044A7CC	E8 2F68FBFF	CALL run.00401000	
0044A7D1	C745 F8 00000000	MOV DWORD PTR SS:[EBP-8],0	
0044A7D8	68 B8C14400	PUSH run.0044C1B8	ASCII "rb"
0044A7DD	68 BCC14400	PUSH run.0044C1BC	ASCII "file"
0044A7E2	FF15 ACC04400	CALL DWORD PTR DS:[44C0AC1]	MSVCR100.fopen
0044A7E8	83C4 08	ADD ESP,8	
0044A7EB	8945 FC	MOV DWORD PTR SS:[EBP-4],EAX	
0044A7EE	E8 0D68FBFF	CALL run.00401000	
0044A7F3	837D FC 00	CMP DWORD PTR SS:[EBP-4],0	
0044A7F7	75 20	JNZ SHORT run.0044A819	
0044A7F9	E8 0268FBFF	CALL run.00401000	
0044A7FE	68 C4C14400	PUSH run.0044C1C4	
0044A803	FF15 B0C04400	CALL DWORD PTR DS:[44C0B01]	MSVCR100.printf
0044A809	83C4 04	ADD ESP,4	
0044A80C	E8 EF67FBFF	CALL run.00401000	
0044A811	6A 00	PUSH 0	

Registers (FPU)

EAX 0019FF04
ECX 00ECECE0 run.<ModuleEntr
EDX 00ECECE0 run.<ModuleEntr
EBX 00381000
ESP 0019FF84
EBP 0019FF94
ESI 00ECECE0 run.<ModuleEntr
EDI 00ECECE0 run.<ModuleEntr
EIP 0044AC9B run.0044AC9B

C 1 ES 002B 32bit 0(FFFFFFFF)
P 1 CS 0023 32bit 0(FFFFFFFF)
A 0 SS 002B 32bit 0(FFFFFFFF)
Z 0 DS 002B 32bit 0(FFFFFFFF)
S 0 FS 0053 32bit 384000(FF
T 0 GS 002B 32bit 0(FFFFFFFF)
D 0
O 0 LastErr ERROR_SUCCESS (0)
EFL 00000207 (NO,B,NE,BE,NS,
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0

FST 0000 Cond 0 0 0 0 Err
FCW 027F Prec NEAR,53 Mask

Address	Hex dump	Disassembly	Comment
00ECF000	0000	ADD BYTE PTR DS:[EAX],AL	0019FF84 76D38484 RETURN to KERNEL
00ECF002	0000	ADD BYTE PTR DS:[EAX],AL	0019FF88 00381000
00ECF004	0000	ADD BYTE PTR DS:[EAX],AL	0019FF8C 76D38460 KERNEL32.BaseThr
00ECF006	0000	ADD BYTE PTR DS:[EAX],AL	0019FF90 F7EDC82A
00ECF008	04 00	ADD AL,0	0019FF94 0019FFDC
00ECF00A	0000	ADD BYTE PTR DS:[EAX],AL	0019FF98 77AC2EC0 RETURN to ntdll.
00ECF00C	0000	ADD BYTE PTR DS:[EAX],AL	0019FF9C 00381000
00ECF00E	0000	ADD BYTE PTR DS:[EAX],AL	0019FFA0 00000000

아까 oep에서 아래 쪽 말고 위쪽으로 올라오다 보면 scanf함수를 찾을 수 있다. scanf 아랫부분에 브레이크 포인트를 걸어두고 f9(브레이크포인트 걸리기 전 까지 명령 모두실행)을 해준다.

Web & Reversing

The screenshot shows a debugger window with the following components:

- Disassembly Window:** Shows assembly code for a program named 'run.exe'. The code includes instructions like `PUSH EAX`, `POP EAX`, `PUSH EBX`, and `CALL run.00401000`. The address `0044A798` is highlighted.
- Hex Dump Window:** Shows the memory dump for the address `0044A798`. The dump shows the ASCII string `"rb"` and the file path `"file"`.
- Registers Window:** Shows the current state of the CPU registers. The `EAX` register contains `00000001`, and the `EIP` register contains `0044A798`.

The hex dump window shows the following data:

Address	Hex dump	Disassembly	Comment
00ECF000	0000	ADD BYTE PTR DS:[EAX],AL	
00ECF002	0000	ADD BYTE PTR DS:[EAX],AL	
00ECF004	0000	ADD BYTE PTR DS:[EAX],AL	
00ECF006	0000	ADD BYTE PTR DS:[EAX],AL	
00ECF008	04 00	ADD AL,0	
00ECF00A	0000	ADD BYTE PTR DS:[EAX],AL	
00ECF00C	0000	ADD BYTE PTR DS:[EAX],AL	

아무 값이나 key로 입력해주고 엔터를 누르고 우리가 생각한대로 scanf명령다음에 브레이크가 걸려서 멈추게 된다.

Web & Reversing

Address	Hex dump	Disassembly	Comment
0044A897	E8 6467FBFF	CALL run.00401000	
0044A89C	C745 F8 00000000	MOV DWORD PTR SS:[EBP-81],0	
0044A8A3	EB 09	JMP SHORT run.0044A8AE	
0044A8A5	8B45 F8	MOV EAX,DWORD PTR SS:[EBP-81]	
0044A8A8	83C0 01	ADD EAX,1	
0044A8AB	8945 F8	MOV DWORD PTR SS:[EBP-81],EAX	
0044A8AE	8B4D F8	MOV ECX,DWORD PTR SS:[EBP-81]	
0044A8B1	3B4D F0	CMP ECX,DWORD PTR SS:[EBP-101]	
0044A8B4	73 49	JNB SHORT run.0044A8FF	
0044A8B6	8B55 F8	MOV EDX,DWORD PTR SS:[EBP-81]	
0044A8B9	0FBE8A B8155400	MOVSB ECX,BYTE PTR DS:[EDX+5415B81]	
0044A8C0	8B45 F8	MOV EAX,DWORD PTR SS:[EBP-81]	
0044A8C3	33D2	XOR EDX,EDX	
0044A8C5	F775 F4	DIV DWORD PTR SS:[EBP-C1]	
0044A8C8	0FBE92 70D34400	MOVSB EDX,BYTE PTR DS:[EDX+44D3701]	
0044A8CF	33CA	XOR ECX,ECX	
0044A8D1	8B45 F8	MOV EAX,DWORD PTR SS:[EBP-81]	
0044A8D4	8888 B8155400	MOV BYTE PTR DS:[EAX+5415B81],CL	
0044A8DA	E8 2167FBFF	CALL run.00401000	
0044A8DF	8B4D F8	MOV ECX,DWORD PTR SS:[EBP-81]	
0044A8E2	0FBE91 B8155400	MOVSB EDX,BYTE PTR DS:[ECX+5415B81]	
0044A8E9	81F2 FF000000	XOR EDX,0FF	
0044A8EF	8B45 F8	MOV EAX,DWORD PTR SS:[EBP-81]	
0044A8F2	8890 B8155400	MOV BYTE PTR DS:[EAX+5415B81],DL	
0044A8F8	E8 0367FBFF	CALL run.00401000	
0044A8FD	EB A6	JMP SHORT run.0044A8A5	
0044A8FF	8B4D FC	MOV ECX,DWORD PTR SS:[EBP-41]	
0044A902	51	PUSH ECX	
0044A903	FF15 A0C04400	CALL DWORD PTR DS:[44C0A01]	MSVCR100.fclose
0044A909	83C4 04	ADD ESP,4	
0044A90C	E8 EF66FBFF	CALL run.00401000	
0044A911	68 DCC14400	PUSH run.0044C1DC	ASCII "wb"
0044A916	68 E0C14400	PUSH run.0044C1E0	ASCII "file"
0044A91B	FF15 ACC04400	CALL DWORD PTR DS:[44C0AC1]	MSVCR100.fopen
0044A921	83C4 08	ADD ESP,8	
0044A924	8945 EC	MOV DWORD PTR SS:[EBP-14],EAX	
0044A927	E8 D466FBFF	CALL run.00401000	
0044A92C	E8 CF66FBFF	CALL run.00401000	
0044A931	C745 F8 00000000	MOV DWORD PTR SS:[EBP-81],0	
0044A938	EB 09	JMP SHORT run.0044A943	
0044A93A	8B55 F8	MOV EDX,DWORD PTR SS:[EBP-81]	
0044A93D	83C2 01	ADD EDX,1	
0044A940	8955 F8	MOV DWORD PTR SS:[EBP-81],EDX	
0044A943	8B45 F8	MOV EAX,DWORD PTR SS:[EBP-81]	

fopen 함수를 통해 'file' 읽어드린후에 pe 바이너리를 암호화한다.
xor ecx,edx xor edx,000000ff 순서로 연산하는데 이를 해석해보면
(originalData[i]^key)^0xff=암호화 된 데이터[i]/i는 문자열 인덱스가 될 수
있겠다. 따라서 키값에 대한 식으로 다시 바꾸면 (암호화 된 데이터[i] ^ key)
^ 0xff=originalData[i]라고 할 수 있다.

Web & Reversing

HxD - [C:\Users\성준\Desktop\리버싱기초\이해\ransomware\run.exe]

파일(F) 편집(E) 찾기(S) 보기(V) 분석(A) 기타 설정(X) 창 설정(W) ?

16 ANSI 16 진수

readme.txt run.exe

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ.....ÿÿ..
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00@.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	F0	00	00	00ð.....
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	...°.i!..Li!Th
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6E	is program cannot
00000060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	be run in DOS
00000070	6D	6F	64	65	2E	0D	0A	24	00	00	00	00	00	00	00	00	mode....\$......
00000080	B8	71	E4	19	FC	10	8A	4A	FC	10	8A	4A	FC	10	8A	4A	..qâ.ü.ŠJü.ŠJü.ŠJ
00000090	6F	5E	12	4A	FD	10	8A	4A	93	66	16	4A	FE	10	8A	4A	o^Jý.ŠJ^f.Jp.ŠJ
000000A0	93	66	14	4A	FD	10	8A	4A	93	66	20	4A	EF	10	8A	4A	"f.Jý.ŠJ^f.Ji.ŠJ
000000B0	93	66	21	4A	FE	10	8A	4A	F5	68	19	4A	FF	10	8A	4A	"f!Jp.ŠJõh.Jý.ŠJ
000000C0	FC	10	8B	4A	CF	10	8A	4A	93	66	25	4A	FD	10	8A	4A	ü.<Ji.ŠJ^f%Jý.ŠJ
000000D0	93	66	17	4A	FD	10	8A	4A	52	69	63	68	FC	10	8A	4A	"f.Jý.ŠJRichü.ŠJ
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	50	45	00	00	4C	01	03	00	32	2B	53	4F	00	00	00	00	PE..L...2+SO....
00000100	00	00	00	00	E0	00	03	01	0B	01	0A	00	00	20	00	00â.....
00000110	00	10	00	00	00	C0	AC	00	E0	EC	AC	00	00	D0	AC	00Ä..âi...Ð~.
00000120	00	F0	AC	00	00	40	00	00	10	00	00	00	00	02	00	00	..ð...@.....
00000130	05	00	01	00	00	00	00	00	05	00	01	00	00	00	00	00
00000140	00	00	AD	00	00	10	00	00	00	00	00	00	00	03	00	40@.
00000150	00	00	10	00	00	10	00	00	00	00	10	00	00	10	00	00
00000160	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00
00000170	B8	F1	AC	00	DC	00	00	00	00	F0	AC	00	B8	01	00	00	..â..Ü....ð~....
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001B0	00	00	00	00	00	00	00	00	88	EE	AC	00	48	00	00	00^i~.H...
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001E0	00	00	00	00	00	00	00	55	50	58	30	00	00	00	00	00UPX0....
000001F0	00	C0	AC	00	00	10	00	00	00	00	00	00	00	04	00	00	..Ä.....
00000200	00	00	00	00	00	00	00	00	00	00	00	00	00	80	00	E0€.â
00000210	55	50	58	31	00	00	00	00	20	00	00	00	D0	AC	00	00	UPX1.....Ð~.
00000220	00	20	00	00	00	04	00	00	00	00	00	00	00	00	00	00
00000230	00	00	00	00	40	00	00	E0	2E	72	73	72	63	00	00	00@..â.rsrc...
00000240	00	10	00	00	F0	AC	00	00	04	00	00	00	24	00	00	00ð.....\$.
00000250	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	C0@..Ä
00000260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000270	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000290	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

오프셋: 4E

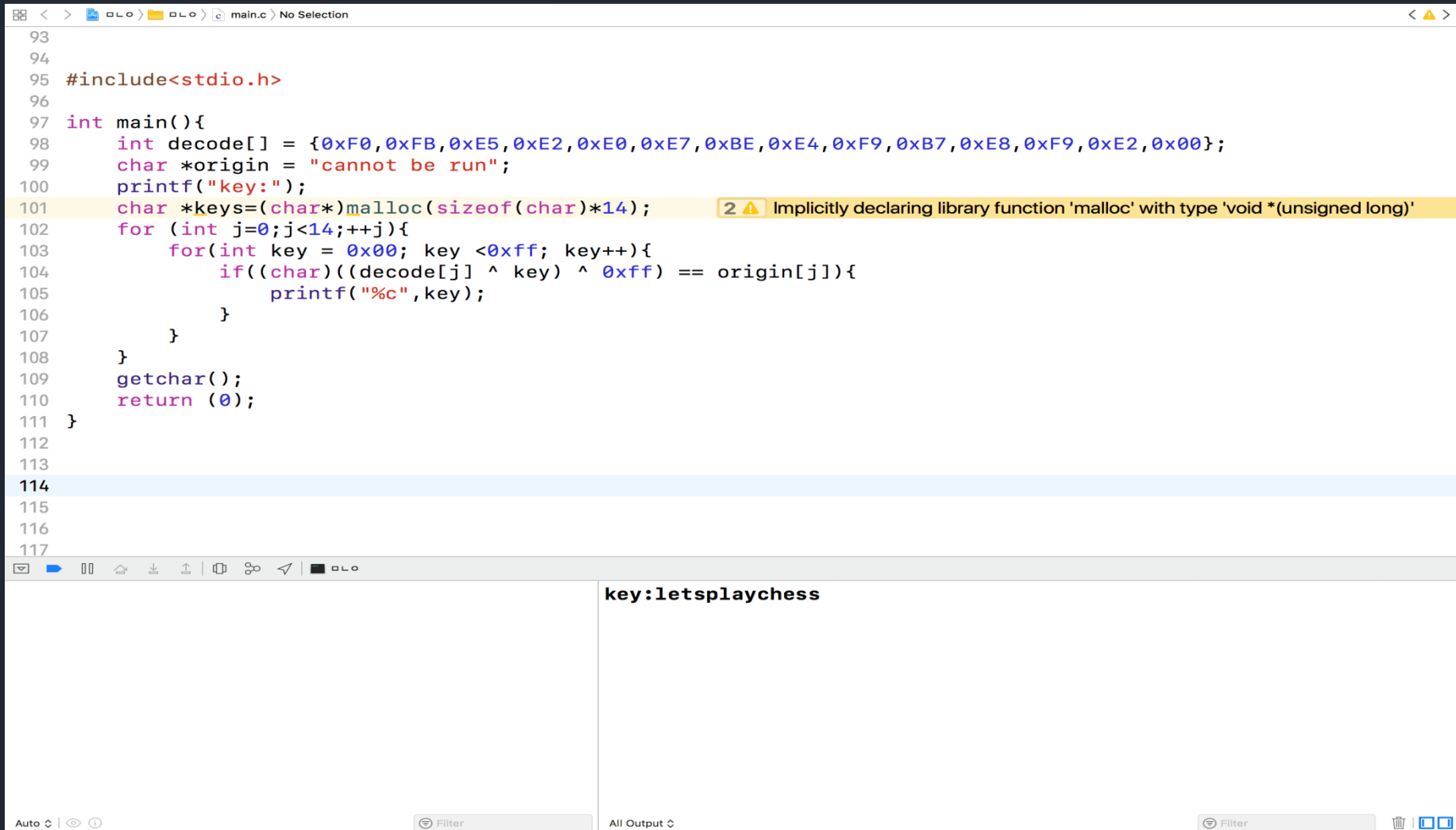
블록: 4E-6F

길이: 22

읽어쓰기

hxd통해 run.exe의 pe 바이너리를 보면 'This is program cannot be run in DOS'라는 문구를 볼 수 있다. 이는 exe파일의 가장 큰 특징이다.

Web & Reversing



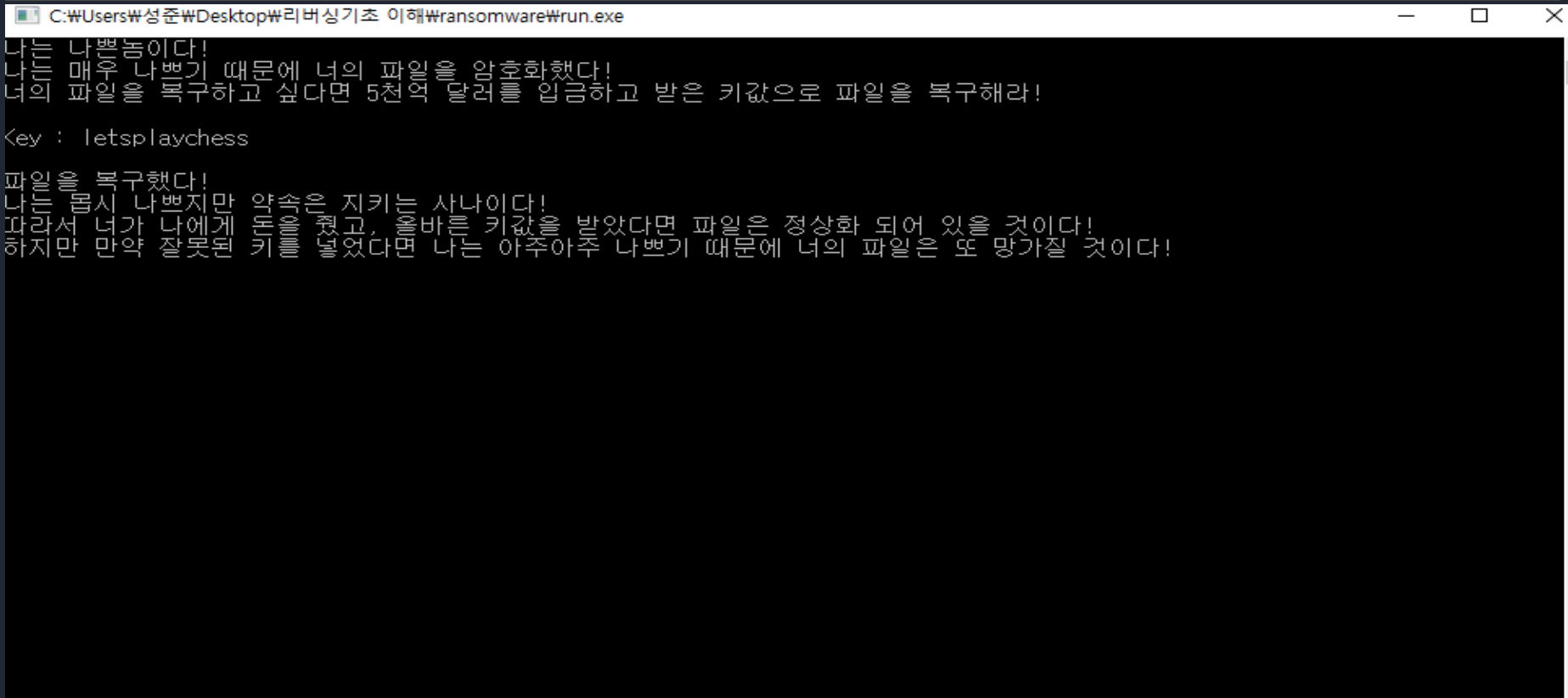
```
93
94
95 #include<stdio.h>
96
97 int main(){
98     int decode[] = {0xF0,0xFB,0xE5,0xE2,0xE0,0xE7,0xBE,0xE4,0xF9,0xB7,0xE8,0xF9,0xE2,0x00};
99     char *origin = "cannot be run";
100    printf("key:");
101    char *keys=(char*)malloc(sizeof(char)*14);
102    for (int j=0;j<14;++j){
103        for(int key = 0x00; key <0xff; key++){
104            if((char)((decode[j] ^ key) ^ 0xff) == origin[j]){
105                printf("%c",key);
106            }
107        }
108    }
109    getchar();
110    return (0);
111 }
112
113
114
115
116
117
```

2 ⚠ Implicitly declaring library function 'malloc' with type 'void *(unsigned long)'

key:letsplaychess

키 값이 'letsplaychess'란걸 알 수 있다.
한번 run에 집어 넣어서 file을 복구시키자.

Web & Reversing



키 값을 입력하면 복호화를 진행해준다.

Web & Reversing

HxD - [C:\Users\정준\Desktop\리버싱기초 이해\ransomware\file]

파일(F) 편집(E) 찾기(S) 보기(V) 분석(A) 기타 설정(X) 창 설정(W) ?

16 ANSI 16 진수

file

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ.....yy..
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00@.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	F0	00	00	008...
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	..°...í!..Lí!Th
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is program canno
00000060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t be run in DOS
00000070	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	mode....\$.
00000080	0D	41	A4	90	49	20	CA	C3	49	20	CA	C3	49	20	CA	C3	.Aw.I ÉÄI ÉÄI ÉÄ
00000090	DA	6E	52	C3	48	20	CA	C3	26	56	56	C3	4A	20	CA	C3	ÚnRÄH ÉÄ&vVÄJ ÉÄ
000000A0	26	56	60	C3	5B	20	CA	C3	40	58	59	C3	4B	20	CA	C3	sV Ä[ÉÄ&XYÄK ÉÄ
000000B0	49	20	CB	C3	71	20	CA	C3	26	56	61	C3	40	20	CA	C3	I ÉÄq ÉÄ&VaÄ@ ÉÄ
000000C0	26	56	50	C3	48	20	CA	C3	26	56	57	C3	48	20	CA	C3	sVPÄH ÉÄ&VWÄH ÉÄ
000000D0	52	69	63	68	49	20	CA	C3	00	00	00	00	00	00	00	00	RichI ÉÄ.....
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	50	45	00	00	4C	01	03	00	BC	2B	53	4F	00	00	00	00	PE...L...4+SO....
00000100	00	00	00	00	E0	00	03	01	0B	01	0A	00	00	20	00	00à.....
00000110	00	10	00	00	00	A0	01	00	B0	C9	01	00	00	B0	01	00°É.....°
00000120	00	D0	01	00	00	00	40	00	00	10	00	00	00	02	00	00	.Ð....@.....
00000130	05	00	01	00	00	00	00	00	05	00	01	00	00	00	00	00
00000140	00	E0	01	00	00	10	00	00	00	00	00	00	03	00	40	81	.à.....@.
00000150	00	00	10	00	00	10	00	00	00	00	10	00	00	10	00	00
00000160	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00
00000170	F4	D1	01	00	DC	00	00	00	00	D0	01	00	F4	01	00	00	ôÑ...Ü....Ð...ô...
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001E0	00	00	00	00	00	00	00	00	55	50	58	30	00	00	00	00UPX0....
000001F0	00	A0	01	00	00	10	00	00	00	00	00	00	00	04	00	00
00000200	00	00	00	00	00	00	00	00	00	00	00	00	80	00	00	E0€...à
00000210	55	50	58	31	00	00	00	00	20	00	00	00	00	B0	01	00	UPX1.....°..
00000220	00	1C	00	00	00	04	00	00	00	00	00	00	00	00	00	00
00000230	00	00	00	00	40	00	00	E0	2E	72	73	72	63	00	00	00@...à.rsrc...
00000240	00	10	00	00	00	D0	01	00	00	04	00	00	00	20	00	00Ð.....
00000250	00	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00@...Ä
00000260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000270	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

hxd로보면 pe헤더가 mz로 시작한다 올바르게 복호화 된 것이다.
이제 확장자를 .exe로 바꾸고 실행 해보자.

file.exe

2017-07-12 오후 5:36

응용 프로그램

9KB

복호화 됐으니 exe파일로 바꿔서 실행을 시켜보자.

C:\Users\성준\Desktop\리버싱기초 이해\ransomware\file.exe

Key -> Colle System

키 값이 나왔다.

The end
