

InTouch v9.5 license cracking by gerbay (from Türkiye)... For FlexLM 5.X protection.. **Only for education, other usage prohibited..**

Important functions:

```
int lc_init(
    LM_HANDLE *    oldjob,      /* OLD license job */
    char *         vendor_id,   /* vendor ID */
    VENDORCODE *   vendor_key, /* Vendor's encryption seeds */
    LM_HANDLE **   job_id
);

int lc_checkout(
    LM_HANDLE_PTR    job,      /* Current license job */
    const LM_CHAR_PTR feature, /* The feature to be checked in */
    const LM_CHAR_PTR version, /* Feature's version */
    int              nlic,     /* Number of licenses to checkout */
    int              flag,     /* Checkout flag */
    const VENDORCODE_PTR key,   /* The vendor's key */
    int              dup_group  /* Duplicate license grouping criteria */
);

unsigned long l_svk(
    char *         vendor_id,
    VENDORCODE *   key /*- l_svk means "signature vendor_key5" */
);
```

InTouch v9.5 uses FlexLM 5.0a license protection (**Imgr325a.dll**)

We can use **IDA** (Interactive Disassembler) or **OillyDbg**. I use both of them.
Also we need **Imkg.exe** for vendor key generation..

In Imgr325a.dll

Method Name	Is Exported	Address	Importance
lc_init	Exported Entry 52	Base + 0x00D850	VENDOR_NAME and Vendor's Key
lc_checkout	Exported Entry 34	Base + 0x009550	FEATURE name, version, Vendor's Key
l_svk	Local Function, not public	Base + 0x00A8C0	!!! VENDOR_KEY5 !!!

VENDORCODE structure defined:

```
typedef struct vendorcode
{
    short type;
    unsigned long data[2];
    unsigned long keys[4];
    short flexlm_version;
    short flexlm_revision;
};
```

vendorcode.type defined short (2 byte) but, for the compiler's packing alignment, It's size is 4 byte..

So;

```
vendorcode.data[0] = vendorcode_PTR + 4,
vendorcode.data[1] = vendorcode_PTR + 8,
```

```
vendorcode.data[0] = ENCRYPTION_SEED1 ^ VENDOR_KEY5;
vendorcode.data[1] = ENCRYPTION_SEED2 ^ VENDOR_KEY5;
```

Now, we are starting;

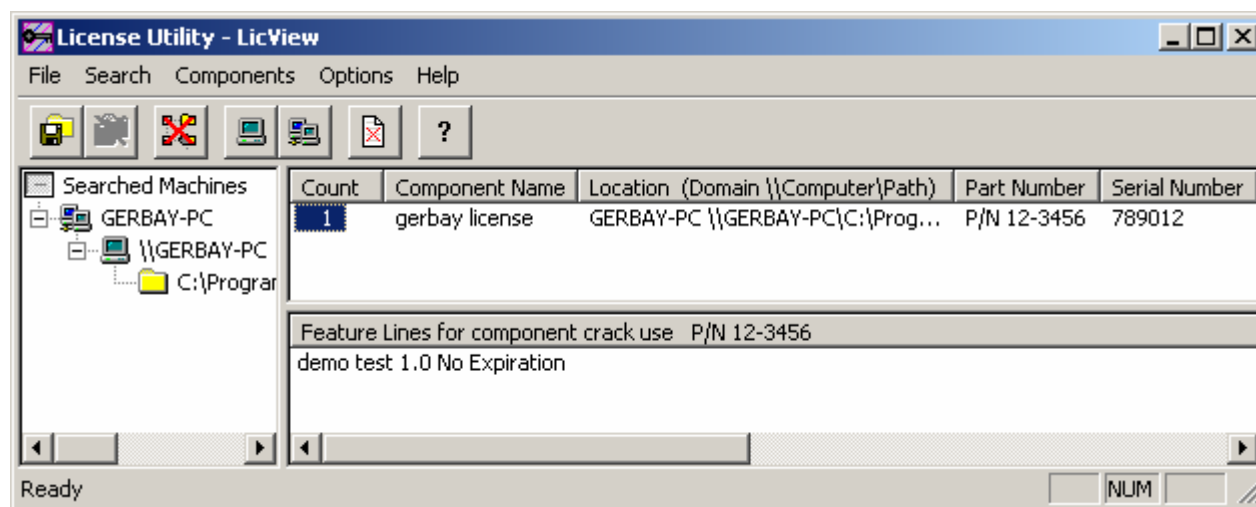
You can read <http://www.icsen.com/technotes/html/overvie2.htm> page for overview of the factory suite 2000 license system.. license file format shown there.. we should prepare dummy license file like shown below:

--- WWSuite.lic --- in "C:\Program Files\Common Files\ArchestrA\License" directory ---

```
#gerbay license|crack use|P/N 12-3456|#Serial Number 789012|demo
FEATURE demo test 1.0 01-Jan-00 0 0 \
  VENDOR_STRING=gerbay HOSTID=ANY \
  ISSUER="gerbay" \
  NOTICE="gerbay" SN=789012
```

--- WWSuite.lic --- end -

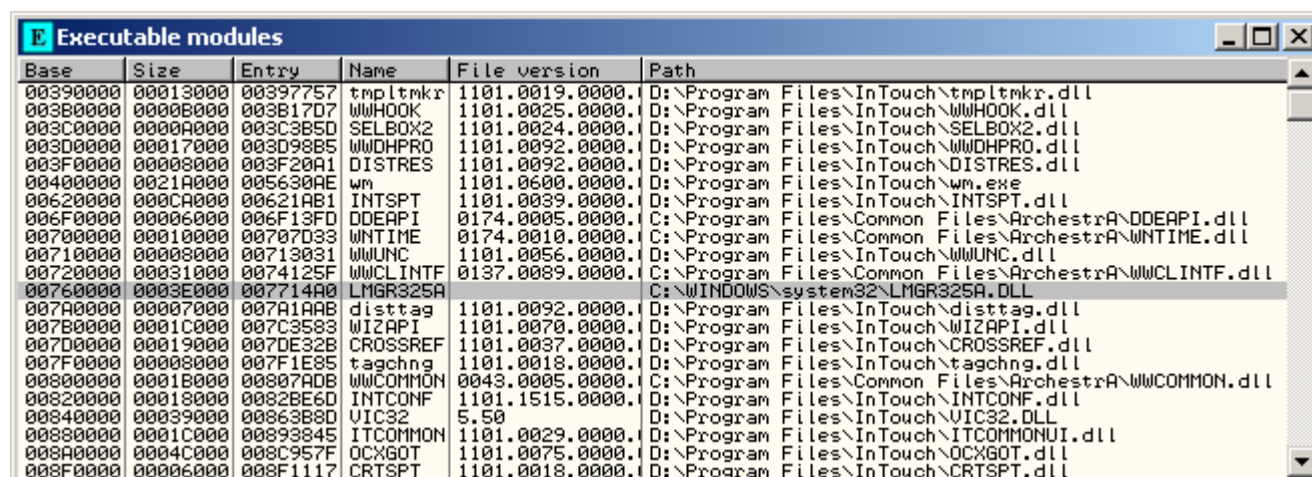
Run InTouch's license utility and see below:



BUT it is not a real license!!!

NOW we can start for recovering license data and recovering encryption seeds.

- run ollydbg.exe and load wm.exe in InTouch package..
- select "Executable Modules" in "View" menu



Find loaded lmgr325a.dll module and than click it.

```

CPU - main thread, module LMGR325A
00761000 1E      PUSH DS
00761001 06      PUSH ES
00761002 55      PUSH EBP
00761003 56      PUSH ESI
00761004 57      PUSH EDI
00761005 52      PUSH EDX
00761006 51      PUSH ECX
00761007 53      PUSH EBX
00761008 50      PUSH EAX
00761009 A3 70C67800 MOV DWORD PTR DS:[78C670],EAX
0076100E 891D 74C67800 MOV DWORD PTR DS:[78C674],EBX
00761014 890D 78C67800 MOV DWORD PTR DS:[78C678],ECX
0076101A E8 83040000 CALL LMGR325A.007614A2
0076101F E8 D9050000 CALL LMGR325A.007615FD
00761024 A1 70C67800 MOV EAX,DWORD PTR DS:[78C670]
00761029 25 FF1F0000 AND EAX,1FFF
0076102E 75 05      JNZ SHORT LMGR325A.00761035
00761030 A1 64C67800 MOV EAX,DWORD PTR DS:[78C664]
00761035 A3 58C67800 MOV DWORD PTR DS:[78C658],EAX
0076103A 8B1D 74C67800 MOV EBX,DWORD PTR DS:[78C674]
00761040 81E3 FF1F0000 AND EBX,1FFF
00761046 75 06      JNZ SHORT LMGR325A.0076104E
00761048 8B1D 68C67800 MOV EBX,DWORD PTR DS:[78C668]
0076104E 891D 5CC67800 MOV DWORD PTR DS:[78C65C],EBX
00761054 8B0D 78C67800 MOV ECX,DWORD PTR DS:[78C678]
0076105A 81E1 FF1F0000 AND ECX,1FFF
00761060 75 06      JNZ SHORT LMGR325A.00761068

```

(module base address is 0x761000)

Find our important methods and set to breakpoint..

lc_init = 0x761000 + 0x00D850 = 0x76E850

lc_checkout = 0x761000 + 0x009550 = 0x76A550

l_svk = 0x761000 + 0x00A8C0 = 0x76B8C0

for example, breakpoint for the lc_init method...

```

CPU - main thread, module LMGR325A
0076E850 81EC A0010000 SUB ESP,1A0
0076E856 C74424 08 010000 MOV DWORD PTR SS:[ESP+8],1
0076E85E 53      PUSH EBX
0076E85F 56      PUSH ESI
0076E860 57      PUSH EDI
0076E861 33F6    XOR ESI,ESI
0076E863 897424 18 MOV DWORD PTR SS:[ESP+18],ESI
0076E867 55      PUSH EBP

```

Our breakpoints shown below:

Breakpoints			
Address	Module	Active	Disassembly
005630AE	wn	One-shot	PUSH EBP
0076A550	LMGR325A	Always	PUSH ESI
0076B8C0	LMGR325A	Always	MOV EAX,DWORD PTR SS:[ESP+8]
0076E850	LMGR325A	Always	SUB ESP,1A0

Now.. pres F9 button in Ollydbg..

If any exception occurred try to debugging.. (you can control breakpoint list)..

if Debugger reaches lc_init:

The screenshot shows a debugger window titled "CPU - main thread, module LMGR325A". It is divided into three main panes:

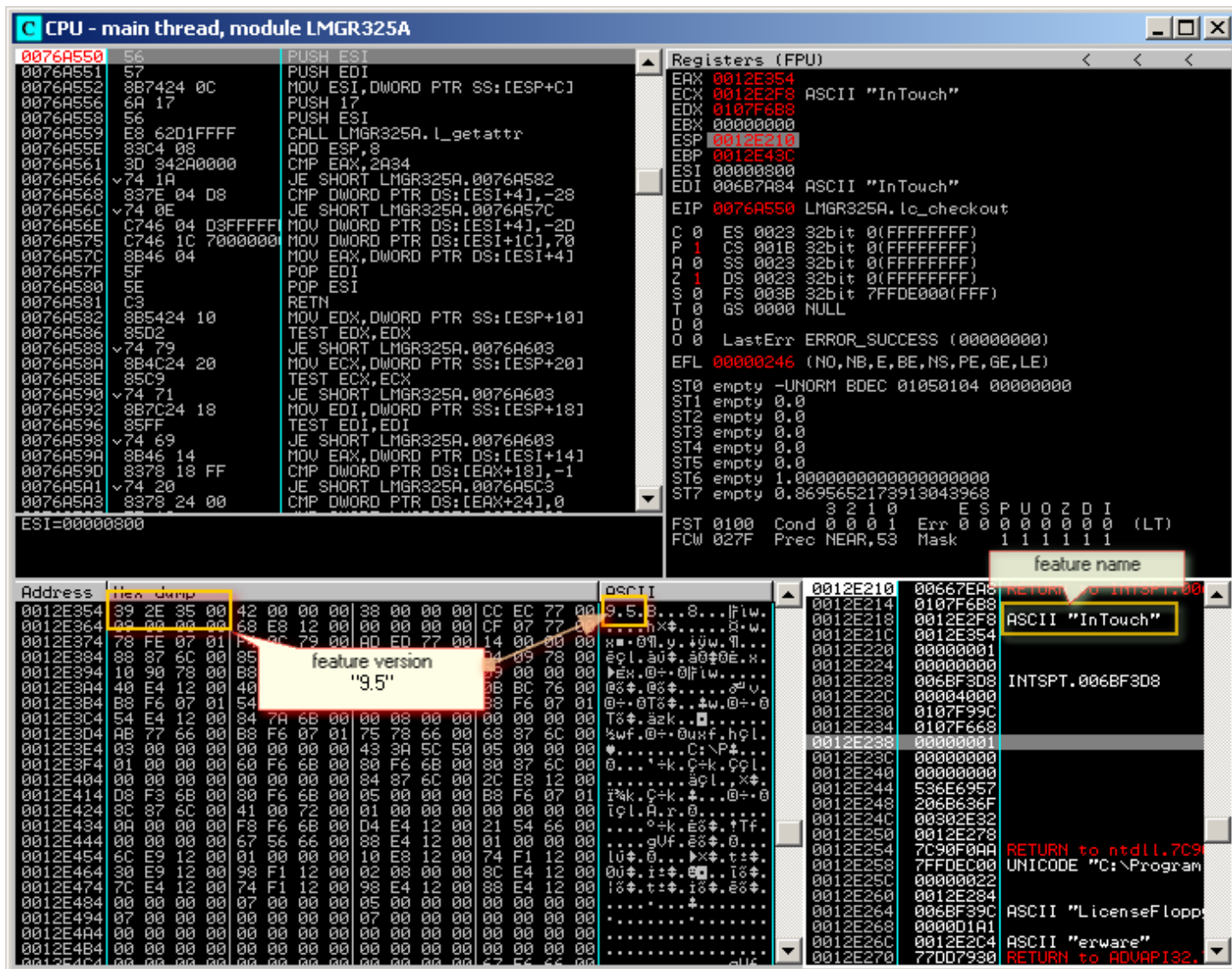
- Assembly Pane (Left):** Displays assembly instructions with their addresses. The instruction at address 0076E850 is highlighted: `81EC A0010000 SUB ESP,1A0`. Other instructions include `INT3`, `MOV DWORD PTR SS:[ESP+8],1`, `PUSH EBX`, `PUSH ESI`, `PUSH EDI`, `XOR ESI,ESI`, `MOV DWORD PTR SS:[ESP+18],ESI`, `PUSH EBP`, `MOV DWORD PTR SS:[ESP+10],ESI`, `CMP DWORD PTR SS:[ESP+1B4],ESI`, `JE SHORT LMGR325A.0076E8C2`, `MOV EAX,DWORD PTR SS:[ESP+1B4]`, `PUSH ID`, `PUSH EAX`, `CALL LMGR325A.L_getattr`, `ADD ESP,8`, `CMP EAX,9969`, `JE SHORT LMGR325A.0076E8C2`, `MOV EAX,DWORD PTR SS:[ESP+1B4]`, `CMP DWORD PTR DS:[EAX],67`, `JE SHORT LMGR325A.0076E8C2`, `CMP EAX,ESI`, `JE SHORT LMGR325A.0076E8B2`, `CMP DWORD PTR DS:[EAX+1],-28`, and `JE SHORT LMGR325A.0076E8B2`.
- Registers (FPU) Pane (Top Right):** Shows the state of CPU registers. `EAX` contains `0068FAE8` (ASCII "Wonderware"). `EDX` contains `0012E420`. `EBX` contains `006C23F8` (INTSPT.006C23F8). `ESI` contains `00000300`. `EDI` contains `006B7A84` (ASCII "InTouch"). `EIP` points to `0076E850` (LMGR325A.Lc_init).
- Memory Dump Pane (Bottom):** Shows a hex dump of memory starting at address 006BF3D8. A yellow box highlights the "vendorcode structure" at address 006BF3F8, which contains the following data: `04 00 00 00`, `1C 86 DE 8E`, `65 CC 9A DE`, `3D B7 65 F2`, `7C B3 3A F5`, `7E 60 DA FC`, `DF FE DF BC`, `05 00 00 00`. The ASCII column shows the string "Wonderware" starting at offset 006BF3E8.

vendor_id = "Wonderware" (case sensitive)

vendorcode structure (encrypted):
 type = 0x00000004 (4 byte)
 data[0] = 0x8EDE861C (4 byte)
 data[1] = 0xDE9ACC65 (4 byte)
 keys[0] = 0xF265B73D (4 byte)
 keys[1] = 0xF53AB37C (4 byte)
 keys[2] = 0xFCDA607E (4 byte)
 keys[3] = 0xBCDFFEDF (4 byte)
 flexlm_version = 0x0005 (2 byte)
 flexlm_revision = 0x0000 (2 byte)

continue to debug...

if debugger reaches Lc_checkout:



Feature name = "InTouch"

Version string = "9.5"

And continue to debug,

New features detected for each `Lc_checkout` calls..

Now we can change license file.. new license file structure shown below:

```
#gerbay license|crack use|P/N 12-3456|#Serial Number 789012|InTouch
FEATURE InTouch Wonderware 9.5 01-Jan-00 0 0 \
VENDOR_STRING=gerbay HOSTID=ANY \
ISSUER="gerbay" NOTICE="gerbay" SN=789012
```

Now,

we can load `InTouch.exe` to `ollydbg` for recovering encrypted seeds..

I load `InTocuh.exe` and `lmgr325a.dll` loaded base address: `0x00561000`

we put the breakpoint at `L_svk` method (`0x0056B8C0`)..

if debugger reached `L_svk` method:

CPU - main thread, module LMGR325A

0056B8C0 8B4424 08 MOV EAX,DWORD PTR SS:[ESP+8]
 0056B8C4 83EC 04 SUB ESP,4
 0056B8C7 83C0 0C ADD EAX,0C
 0056B8CA 56 PUSH ESI
 0056B8CB 8B7424 0C MOV ESI,DWORD PTR SS:[ESP+C]
 0056B8CF 50 PUSH EAX
 0056B8D0 56 PUSH ESI
 0056B8D1 E8 4ACFFFF CALL LMGR325A.0056B8520
 0056B8D6 83C4 08 ADD ESP,8
 0056B8D9 85C0 TEST EAX,EAX
 0056B8DB JNZ SHORT LMGR325A.0056B8E4
 0056B8DD 33C0 XOR EAX,EAX
 0056B8DF 5E POP ESI
 0056B8E0 83C4 04 ADD ESP,4
 0056B8E3 C3 RETN
 0056B8E4 B9 03000000 MOV ECX,3
 0056B8E9 33D2 XOR EDX,EDX
 0056B8EB 8B5424 07 MOV BYTE PTR SS:[ESP+7],DL
 0056B8EF 8B5424 06 MOV BYTE PTR SS:[ESP+6],DL
 0056B8F3 8B5424 05 MOV BYTE PTR SS:[ESP+5],DL
 0056B8F7 8B5424 04 MOV BYTE PTR SS:[ESP+4],DL
 0056B8FB 3B16 CMP BYTE PTR DS:[ESI],DL
 0056B8FD JNE SHORT LMGR325A.0056B913
 0056B8FF 8A16 MOV DL,BYTE PTR DS:[ESI]
 0056B901 46 INC ESI
 0056B902 30540C 04 XOR BYTE PTR SS:[ESP+ECX+4],DL
 0056B906 49 DEC ECX
 0056B907 JNS SHORT LMGR325A.0056B90E
 0056B909 B9 03000000 MOV ECX,3
 0056B90E 803E 00 CMP BYTE PTR DS:[ESI],0
 0056B911 JNZ SHORT LMGR325A.0056B8FF
 0056B913 0FB5424 07 MOVSDX EDX,BYTE PTR SS:[ESP+7]
 0056B915 8B5424 06 MOV BYTE PTR SS:[ESP+6],DL
 0056B917 8B5424 05 MOV BYTE PTR SS:[ESP+5],DL
 0056B919 8B5424 04 MOV BYTE PTR SS:[ESP+4],DL
 0056B91B 3B16 CMP BYTE PTR DS:[ESI],DL
 0056B91D JNE SHORT LMGR325A.0056B913

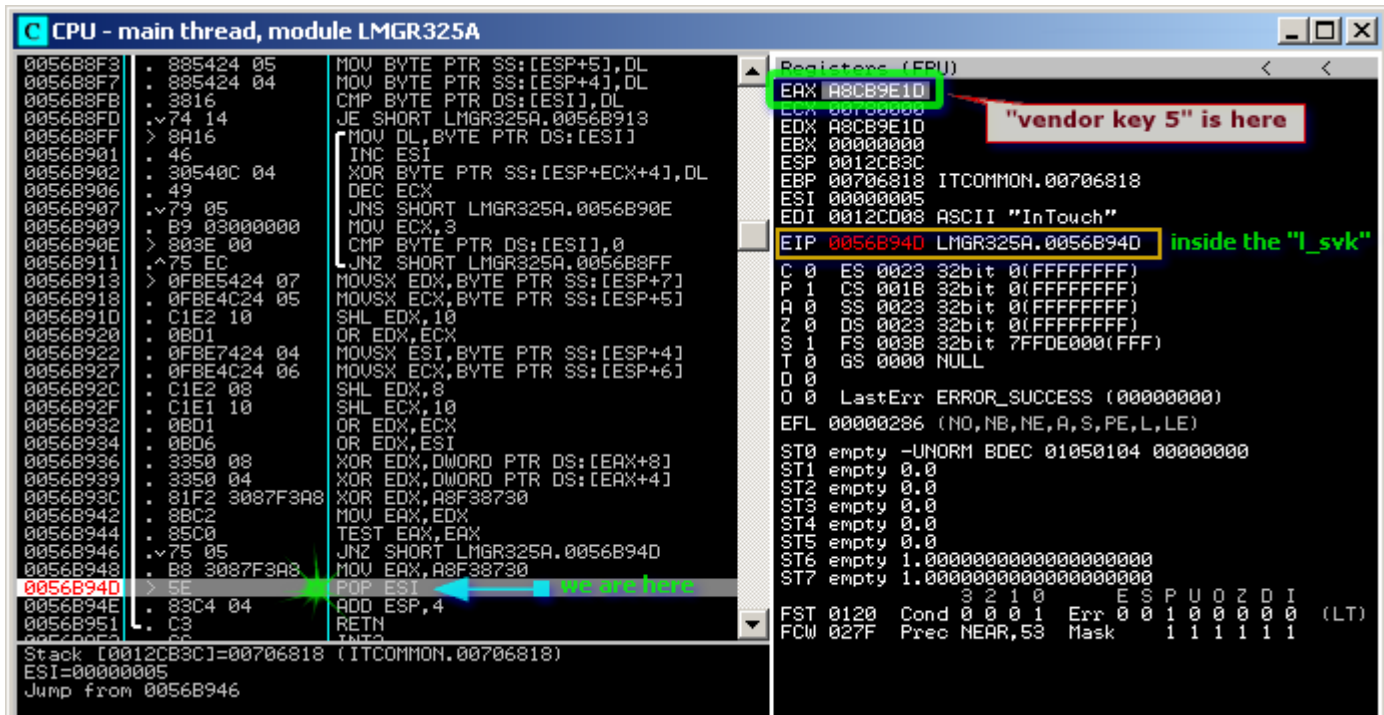
Registers (FPU)
 EAX 00D10B30 ASCII "Wonderware"
 ECX 00D10B30
 EDI 00D10B30
 EBX 00000000
 ESP 0012CB44
 EBP 00706818 ITCOMMON.00706818
 ESI 00706818 ITCOMMON.00706818
 EDI 0012C008 ASCII "InTouch"
 EIP 0056B8C0 LMGR325A.0056B8C0
 C 0 ES 0023 32bit 0(FFFFFFFF)
 P 1 CS 001B 32bit 0(FFFFFFFF)
 A 0 SS 0023 32bit 0(FFFFFFFF)
 Z 0 DS 0023 32bit 0(FFFFFFFF)
 S 0 FS 003B 32bit 7FDE000(FFF)
 T 0 GS 0000 NULL
 D 0
 I 0
 O 0 LastErr ERROR_SUCCESS (00000000)
 EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)
 ST0 empty -UNORM BDEC 01050104 00000000
 ST1 empty 0.0
 ST2 empty 0.0
 ST3 empty 0.0
 ST4 empty 0.0
 ST5 empty 0.0
 ST6 empty 1.000000000000000000000000
 ST7 empty 1.000000000000000000000000
 3 2 1 0 E S P U O Z D I
 FST 0120 Cond 0 0 0 1 Err 0 0 1 0 0 0 0 (LT)
 FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

Stack SS:[0012CB4C]=00706818 (ITCOMMON.00706818)
 EAX=00D10B30, (ASCII "Wonderware")
 Local calls from 0056B8CB, 00567A49, 0056ACA0, 0056AD5D, lc_fes

Address	Hex dump	ASCII
00417000	00 00 00 00 2F F2 40 00 E2 94 40 00 2E B1 40 00/ @.00e...@e.
00417010	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	S.O.F.T.W.A.R.E.
00417020	53 00 4F 00 46 00 54 00 57 00 41 00 52 00 45 00	\.A.R.c.h.e.s.t.
00417030	5C 00 41 00 72 00 63 00 68 00 65 00 73 00 74 00	r.A.\.F.r.a.m.e.
00417040	72 00 41 00 5C 00 46 00 72 00 61 00 60 00 65 00	w.o.r.k.\.A.d.m.
00417050	77 00 6F 00 72 00 68 00 5C 00 41 00 64 00 60 00	i.n.U.s.e.r....
00417060	69 00 6E 00 55 00 73 00 65 00 72 00 00 00 00 00	I.n.s.t.a.l.l.P.
00417070	49 00 6E 00 73 00 74 00 61 00 6C 00 6C 00 50 00	a.t.h...GetAdmin
00417080	61 00 74 00 68 00 00 00 47 65 74 41 64 6D 69 6E	AccountInfo.A.d.
00417090	41 63 63 6F 75 6E 74 49 6E 66 6F 00 41 00 64 00	m.i.n.A.c.o.o.u.
004170A0	6D 00 69 00 6E 00 41 00 63 00 63 00 6F 00 75 00	n.t.I.n.f.o.o.d.
004170B0	6E 00 74 00 49 00 6E 00 66 00 6F 00 2E 00 64 00	l.l....GetAdmin
004170C0	6C 00 6C 00 00 00 00 00 47 65 74 41 64 6D 69 6E	AccountInfoEx...
004170D0	41 63 63 6F 75 6E 74 49 6E 66 6F 45 78 00 00 00	RegisterAndSaveA
004170E0	52 65 67 69 73 74 65 72 41 6E 64 53 61 76 65 41	ppID...@...into
004170F0	70 70 49 44 00 00 00 00 01 00 00 00 69 6E 74 6F	uch.ini.*.*.\...
00417100	75 63 68 2E 69 6E 69 00 2A 2E 2A 00 5C 00 00 00	CFlatToolBar...
00417110	43 46 6C 61 74 54 6F 6F 6C 42 61 72 00 00 00 00	Wonderware.GuiTw
00417120	57 6F 6E 64 65 72 77 61 72 65 2E 47 75 69 54 77	eaks....CloseThe
00417130	65 61 68 73 00 00 00 00 43 6C 6F 73 65 54 68 65	meData...DrawThem
00417140	6D 65 44 61 74 61 00 00 44 72 61 77 54 68 65 6D	eParentBackgroun
00417150	65 50 61 72 65 6E 74 42 61 63 68 67 72 6F 75 6E	d...T.o.o.l.B.a.
00417160	64 00 00 00 54 00 6F 00 6F 00 6C 00 42 00 61 00	r...OpenThemeDat
00417170	72 00 00 00 4F 70 65 6E 54 68 65 6D 65 44 61 74	a...UxTheme.dll.
00417180	61 00 00 00 55 78 54 68 65 6D 65 2E 64 6C 6C 00	this is a test...
00417190	74 68 69 73 2A 69 73 2A 61 2A 74 65 73 74 00 00	

Address	Hex dump	ASCII
0012CB44	0056ACA5	RETURN to LMGR325A.
0012CB48	00D10B30	ASCII "Wonderware"
0012CB4C	00706818	ITCOMMON.00706818
0012CB50	00D15508	
0012CB54	0012CD08	ASCII "InTouch"
0012CB58	00D10A00	
0012CB5C	00706818	ITCOMMON.00706818
0012CB60	00000000	
0012CB64	00D10B30	ASCII "Wonderware"
0012CB68	00D10A00	
0012CB6C	00000001	
0012CB70	00567BE5	RETURN to LMGR325A.
0012CB74	00D10A00	
0012CB78	00D154A8	
0012CB7C	00D15508	
0012CB80	00D10A00	
0012CB84	0056ABA0	RETURN to LMGR325A.
0012CB88	00D10A00	
0012CB8C	00D154A8	
0012CB90	0012CD08	ASCII "InTouch"
0012CB94	00D10A00	
0012CB98	00706818	ITCOMMON.00706818
0012CB9C	0056A75A	RETURN to LMGR325A.
0012CBA0	00D10A00	
0012CBA4	0056A812	RETURN to LMGR325A.
0012CBA8	00D10A00	
0012CBAC	00D15508	

Now continue debug with one step execution using F8 function key (step over)...



Now, we have "vendor name", vendorcode structure, "feature name", version string

```
vendorcode.data[0] = 0x8EDE861C
vendorcode.data[1] = 0xDE9ACC65
```

```
vendorcode.data[0] = ENCRYPTION_SEED1 ^ VENDOR_KEY5;
vendorcode.data[1] = ENCRYPTION_SEED2 ^ VENDOR_KEY5;
```

```
ENCRYPTION_SEED1 = vendorcode.data[0] ^ VENDOR_KEY5;
ENCRYPTION_SEED2 = vendorcode.data[1] ^ VENDOR_KEY5;
```

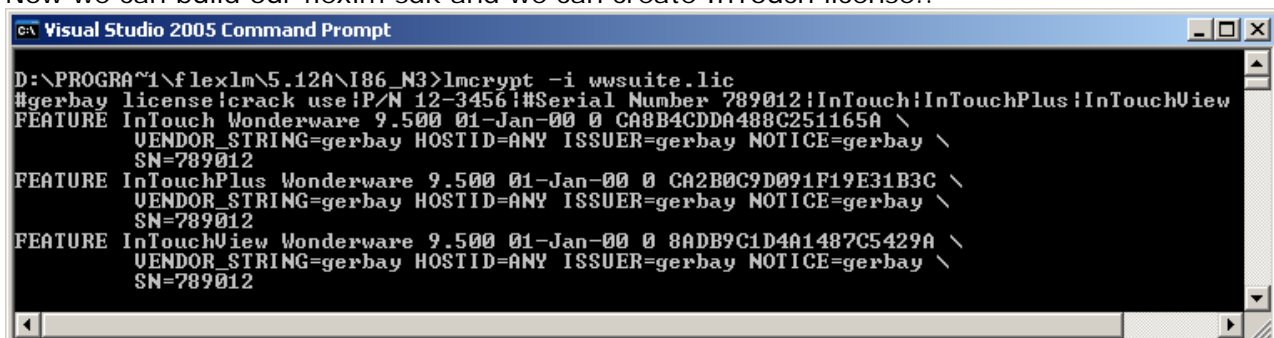
```
ENCRYPTION_SEED1 = 0x8EDE861C ^ 0xA8CB9E1D = 0x26151801
ENCRYPTION_SEED2 = 0xDE9ACC65 ^ 0xA8CB9E1D = 0x76515278
```

We can create vendor keys using lmk.exe (for flexlm version 5.x):

```
/* Version 5 keys */
#define VENDOR_KEY1 0x7a74a41d
#define VENDOR_KEY2 0xe4c5795a
#define VENDOR_KEY3 0x1272b9c7
#define VENDOR_KEY4 0xd87710bf
#define VENDOR_KEY5 0xe88b9e28

#define VENDOR_NAME "Wonderware"
//-- and our detected seeds -----
#define ENCRYPTION_SEED1 0x26151801
#define ENCRYPTION_SEED2 0x76515278
```

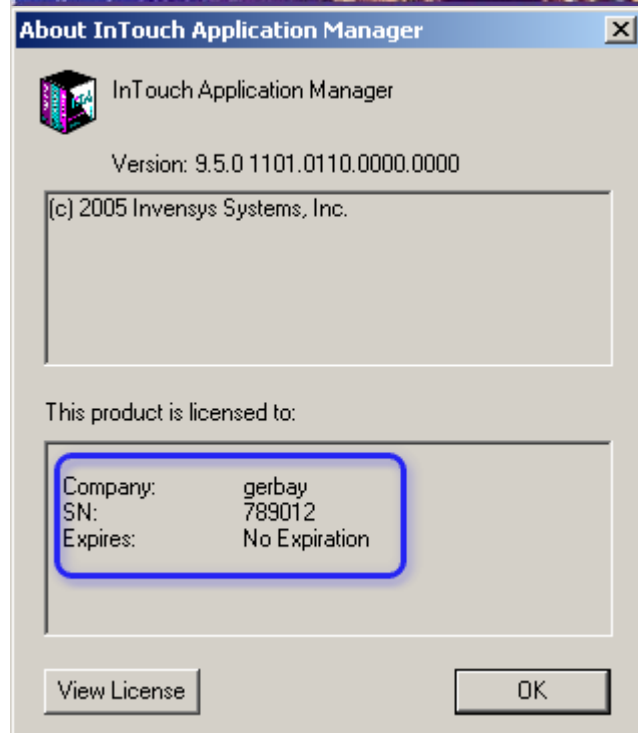
Now we can build our flexlm sdk and we can create InTouch license..



You can copy this license information to wwsuite.lic file:

```
#gerbay license|crack use|P/N 12-3456|#Serial Number 789012|InTouch|InTouchPlus|InTouchView
FEATURE InTouch Wonderware 9.500 01-Jan-00 0 CA8B4CDDA488C251165A \
    VENDOR_STRING=gerbay HOSTID=ANY ISSUER=gerbay NOTICE=gerbay \
    SN=789012
FEATURE InTouchPlus Wonderware 9.500 01-Jan-00 0 CA2B0C9D091F19E31B3C \
    VENDOR_STRING=gerbay HOSTID=ANY ISSUER=gerbay NOTICE=gerbay \
    SN=789012
FEATURE InTouchView Wonderware 9.500 01-Jan-00 0 8ADB9C1D4A1487C5429A \
    VENDOR_STRING=gerbay HOSTID=ANY ISSUER=gerbay NOTICE=gerbay \
    SN=789012
```

And run InTouch.exe program..



Operation completed..

You can detect other FEATURES and version strings and you can create licenses..