

Forum: [Hack Forums](#)

Thread: [\[REQ\] Decrypt an encrypted license](#)

vmtvnn - December 5, 2014, 11:26 pm

General curiosity has the best of me. I'm trying to figure out the license format for this software. I have a valid license for it but I want to know the format of the license file anyways. It seems pretty simple to me but I can't follow all the sub routines. The binary isn't encrypted, and the functions aren't hidden. Everything is out in the open for someone to put together.

Here is the software:

uggc://jjj.znpufhccbeg.pbz/jc-pbagrag/hcybnqf/2013/04/Znpu3lrefvba3.043.066.rkr (Decode with ROT-13)

Extract the main EXE, all the code is in there.

ProtectionID Output:

Quote:Scanning -> C:**********.exe

File Type : 32-Bit Exe (Subsystem : Win GUI / 2), Size : 4456960 (0440200h) Byte(s)

[File Heuristics] -> Flag : 00000010011000000 (0x0004C000)

[Entrypoint Section Entropy] : 6.47

[CompilerDetect] -> Visual C++ 7.1 (Visual Studio 2003)

[!] File appears to have no protection or is using an unknown protection

- Scan Took : 0.344 Second(s) [000000158h tick(s)] [533 scan(s) done]

Here is a valid license in hex format:

Code c6 34 ba 67 0d 0a 7e 27 88 2b 5f d8 a4 96 e5 a1 c0 3d ce d6 72 c0 e6 ac 1c b2 4f 56 f1 d9
bc bf 39 1f 85 a7 9b de c0 f2 8f 72 58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac
ae 13 72 58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac ae
13 72 58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac ae 13
72 58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac ae 13 72
58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac ae 13 72 58 36 a7 f7 ac ae 13 72 58
36 a7 f7 ac ae 13

The first 17 bytes are unchanging. The final 160 bytes are repeating filler. The bytes in between is the plain text license name. It is encrypted in a way where 8 bytes = 6 ASCII characters. Its almost like it is Base64 doubled... but I don't think it is Base64 unless some bytes are transposed since it uses 8 bit (256 characters). There are a few checksum digits mixed in every 8 bytes in the middle. Other than that, there is no verification for a valid license. If the license exists... its considered valid by the program.

The license is saved to the main folder as a .dat file. Search the code for Lic.dat for the full filename and functions where it opens the license file, decodes it and returns it back. "Licensed To" is an important string to search in the file too.

If anyone can help I'd definitely pay some BTC to someone!

(Note I'm trying to keep this obfuscated... so please keep the license filename and program name

out of the thread.).

vmtvnn - December 8, 2014, 12:22 pm

This appears to be the code that takes in the license file and is the first few subroutines. It goes on for a very long time with sub calls. But if anyone can help explain what is going on here.. I'd appreciate it.

For example. What is being done with the DWORDs where it adds the numbers? Is that transposition?

Code // MFC 3.1-10.0 32bit

```
int __thiscall unknown_libname_275(int this)
```

```
{
    int v1; // esi@1
    FILE *v2; // ecx@1
    int result; // eax@1
    int v4; // ecx@3
    char *v5; // esi@4
    int *v6; // eax@5
    void *v7; // ecx@5

    v1 = this;
    v2 = *(FILE **)(this + 16);
    result = 0;
    if ( v2 )
        result = fclose(v2);
    v4 = dword_620908;
    *(_DWORD *)(v1 + 8) = 0;
    *(_DWORD *)(v1 + 16) = 0;
    *(_DWORD *)(v1 + 4) = v4;
    if ( result )
    {
        v5 = *(char **)(v1 + 12);
        v6 = (int *)__doserrno();
        sub_5BFB1E(v7, 13, *v6, v5);
    }
    return result;
}
```

```
void __thiscall sub_5BFB1E(void *this, int a2, int a3, char *a4)
```

```
{
    int v4; // eax@1
    void *v5; // ecx@1
    int v6; // ST0C_4@1
    void *v7; // eax@1
    void *v8; // [sp+4h] [bp-10h]@1
    int v9; // [sp+10h] [bp-4h]@1

    v8 = this;
    v4 = operator new(0x14u);
    v5 = (void *)v4;
    v6 = v4;
    v7 = 0;
    v9 = 0;
```

```

v9 = 0;
if ( v6 )
    v7 = sub_40AAE6(v5, a2, a3, a4);
v9 = -1;
v8 = v7;
_CxxThrowException(&v8, &unk_6457AC);
}

void *__thiscall sub_40AAE6(void *this, int a2, int a3, char *a4)
{
    void *v4; // ST04_4@1

    v4 = this;
    CException::CException();
    *(_DWORD *)v4 = &off_62098C;
    sub_40B0EE((char *)v4 + 16);
    *(_DWORD *)v4 + 2 = a2;
    *(_DWORD *)v4 + 3 = a3;
    sub_40B14B((char *)v4 + 16, a4);
    return v4;
}

```

simpleman24 - December 9, 2014, 9:40 pm

Is this possible if yes can any one teach me....

vmtvnn - December 10, 2014, 12:05 am

I'm learning too. I've been spending time working on it every day looking at the code in IDA, Hexkeys and Olly. I'm still having a heck of a hard time figuring anything out.

I was hoping it would be as easy as debugging through the function and checking what is in memory, but I can't seem to get that working either.

0x22 - December 10, 2014, 2:10 pm

Why bother? Just crack it instead. Just as efficient

vmtvnn - December 11, 2014, 10:16 am

(Dec 10, 2014 03:10 PM)0x22 Wrote: [\[CITE\]](#) Why bother? Just crack it instead. Just as efficient I understand that it would be very easy to just skip over the license check, and force it to think its activated. But I already have a valid license for it. This is mainly just a curiosity. If I wanted to use the software (and didn't have a license), there are plenty of leaked licenses out there too. The software is so old and outdated, they don't even bother blacklisting licenses either. Once you have one.. its valid pretty much forever on any number of computers.

I know that 0x22 is a master as assembly. Would you be able to take a look at it?

I found on their FTP site they have a version dating back to 2003 that is much simpler, and uses the same licensing scheme. It has essentially the same functions as the newest version, but less checking involved.

sgc://nabalzbhf:thrfg@znpuhfccbeg.pbz/Znpu/Znpu3lrefvbaE1.84.002.rkr (Decode link with ROT-13, open as archive and extract main EXE file)

For reference the loading of the license file occurs at: 00461DD2... and I believe the check is at:

0053B1E9 in the above linked download's main EXE.

vmtvnn - *December 24, 2014, 4:20 pm*

The program was just made obsolete, so I'm hoping someone can help figure this out for my own morbid curiosity.

Willing to send some mBTC to whoever can tell me the type of encryption and steps required to decode the license file.