

## Protect your software: tips and recommendations

1. Never create a separated function for checking of registration. Always place the checking code in some important procedure that is required for your program. If cracker will disable this procedure, you program will not work correctly.
2. Never give a meaningful name for procedure that contains a checking (like IsValidSerialNumber, IsTrialExpired or IsValidUser). Change a F2CA name and don't forget about rule#1.
3. If your code returned a message that is not valid serial number or username, then don't try in same moment to show a warning dialog. Save this information somewhere and show it later (better after few days).
4. Have a few different algorithms of validation. I mean that you can generate an activation code using two-three-... algorithms. It allows validating your code in few procedures. If cracker will find a validation procedure then he'll find an one part of validation only. You can check a different part of serial number in other days of month or better to activate this validation after few days after first validation receipt. In this case cracker will "find" a code and publish it but your program will works a few days only. To trace all code of program or spend a lot of time on your program is hard task for any cracker.
5. If you used a some encrypt method for generation of serial number from username, then include a pause in few seconds after username entry before processing. If cracker will use a brute force cracking, this task will take a lot of additional time and in pair with good encrypt method the task of cracking will be unreal.
6. Don't use short keys. Of course, it depends from crypt algorithm but your key must be long. Better to generate a few Kb of key (don't forget to send instruction about key entering) than use a key-generator for 10 characters with standard literal symbols.
7. Don't store algorithm of key generation in your code. It is bad technology when user will type the name and serial number and after that you will run own algorithm for calculation of serial from user name (for next compare). Better to compare some hash and magic structure. Other the cracker will create the own key-generator without any patch of your original exe-file.
8. Store the serial numbers or check sums in different places. Also don't forget that these places must be hidden or with hard access. For example, date/time of file creation, additional invisible line of pixels in bitmap, first

letters of records in database etc. Also don't forget about unique algorithms for encryption of these secure codes – don't use the one algorithm for any hidden places.

9. Don't use the trivial crypt algorithms. The XOR is not good choice. Better to use MD5, RSA, BlowFish, GOST etc
10. If your trial software must be available in 30 days only, then don't try to use a system date-time. Better to use a date-time of some system files (SYSTEM.DAT or DA0, BOOTLOG.TXT, AUTOEXEC.BAT etc). Also don't forget that additionally you can create the some own hidden file during installation and use a date-time of this file.
11. Encrypt the all string resources that will show that program is evaluation or time is expired. You must decrypt it in run-time and/or build dynamically.
12. To lead cracker up the path garden: add the some dummy code with strange external calls in validation procedure. It will increase time of cracking because cracker must found a gold mine in the rubbish.
13. If you're expert in development, try to encrypt the some parts of code in exe/dll-file and decrypt it in run-time. Use a different methods of code checking – checksums, own labels...
14. As I wrote in rule#2 try to use a non-readable names of procedures (D1FA12, B123 etc) or use a names like system functions – GetWindow, MOUSE\_EVENT\_A, KeyboardLayoutW etc. Also you can use the names of some procedures which are popular in your programming language (of course, if this language support the one name for few different procedures in different units/modules).
15. If your programming language supports it, then use the run-time assigning of events. For example, in Borland Delphi/C++Builder you can write a validation code in procedure which will be assigned to OnClick event of some button. It's bad because name of this procedure will be stored in resources. Better to assign a procedure to event in run-time.
16. If you can, don't disable the some extended features in trial version. Better to remove it and after registration customer will receive the other full program (protected, of course).
17. Frequently release a new versions or updates. Of course, don't forget to change the validation method (at least one) in each release.

You must understand that any software can be cracked. And any software will be cracked if your program is good and cost of cracking time is less than your license fee.

If you found that your software cracked, don't panic. In any situation you must find the good news for you. In this situation you must understand that your software is best on market but cracker which broke you protection are better then you now. It is like game – you released a new protection, he must crack it. The next step is your: you must change a protection and add a some hidden surprise for your opponent – new crypt algorithm, new hidden places etc. It's life.

### **Additionally**

Also I want to say the next: the protection is very important part of development process but don't spend a lot of time on it. If you'll concentrate on protection but you'll forget about program possibilities, your software will be not cracked forever. But reason is not a good protection 😊 Your software must worth a time that you spent on protection.

The end-user need a good solution of own problem with good support and new features and your protection must be in background. Exist a lot of protections on hardware keys or new method of confirmation via Internet but these methods will disturb your customers. In this case you can lose them – they will select the other program without any choice for your.

And don't forget about privacy of your customer – you can't select the some personalized data without permissions. No matter that you need it for own protection.