

Forum:[Hack Forums](#)

Thread: [30 day trial, license keys, software authentication question.](#)

dotb1n - December 27, 2017, 12:22 pm

Which one is used the most and which one is the hardest to bypass? I know this is a hacking forum (and you guys have software and tricks for everything) but i guess I want to know which one is the one that gives you all more headaches to crack/break? I am just curious.

Ninja - December 27, 2017, 12:29 pm

A licensing system is what it says; a licensing system. Most are used with a single line of code. Comment out that line of code, and you 'bypassed' it (with the exception of remote variables and all that). Usually they do not offer anti-crack protection, but only things like encrypting network traffic. To protect your code use an obfuscator to 'scramble' your code and make it hard to read.

EZLib - December 27, 2017, 12:44 pm

(Dec 27, 2017 01:22 PM)dotb1n Wrote: [\[CITE\]](#) Which one is used the most and which one is the hardest to bypass? I know this is a hacking forum (and you guys have software and tricks for everything) but i guess I want to know which one is the one that gives you all more headaches to crack/break? I am just curious.

EZLib (a licensing system) has Anti-Debug, and Anti-Dump protection but yet that doesn't mean you're protected. Although it does have its own obfuscator but just not released you can follow more about it here, <https://github.com/developer-variety/EZLib-Issues/issues/5>.

itcorp - December 27, 2017, 2:10 pm

(Dec 27, 2017 01:22 PM)dotb1n Wrote: [\[CITE\]](#) Which one is used the most and which one is the hardest to bypass? I know this is a hacking forum (and you guys have software and tricks for everything) but i guess I want to know which one is the one that gives you all more headaches to crack/break? I am just curious.

As Ninja said, licensing system is what it says; a licensing system comment out that line of code, and you 'bypassed' it .But let's say if you keep business logic on server such as working with databases through some service like wcf or web api, and try to link the whole application to it through it and do some checks like ip or serial number of processors or hardware etc..You may slightly reduce the risk of the application being crack/break but generally nothing is safe.

D?ceived - December 27, 2017, 2:12 pm

Just use the licensing system, then find a obfuscator that works with your program like enigma protector for example

dotb1n - December 28, 2017, 10:00 am

Thanks guys all those are answers that are well taken. I will do research on obfuscators as pointed by you all and try to use them.

Ninja - December 28, 2017, 10:14 am

(Dec 28, 2017 11:00 AM)dotb1n Wrote: [\[CITE\]](#) Thanks guys all those are answers that are well taken. I will do research on obfuscators as pointed by you all and try to use them.

An obfuscator does not license your software though. It does not offer authentication of any kind.

You should use both a licensing system AND an obfuscator. One is to prevent piracy by making

sure only one user with valid credentials can login, and the other is to make sure it does not get cracked.

dotb1n - December 28, 2017, 7:45 pm

I get the difference. the obfuscator piece is what i was missing.

LICENSING:

For my software, I am using a combination of creating a unique number with serial number, mac address number and a few other unique identifiers. This helps me identify a unique machine.

AUTHENTICATION:

Then, i have a web api that authenticates users from the cloud.

PROTECTION(my missing piece):

The piece i was missing was to obfuscate everything so that the code is not crackable.

Remember i am starting and want to protect as much as i can of my intellectual property.

I really appreciate your input and can anyone point me to a post that shows the first 2 (licensing, authentication) items demonstration with a small piece of software? I really want to learn and test my software on my own.

Thanks to all again!

VmX5 - December 28, 2017, 8:01 pm

Obfuscation will help but you want your ONLINE server to do as much as possible and essentially make your app useless without it.

Read what Martin Heidegger said here:

<https://hackforums.net/showthread.php?tid=5774594&pid=56437398#pid56437398>

Aristocles - December 29, 2017, 7:51 am

(Dec 28, 2017 08:45 PM)dotb1n Wrote: [\[CITE\]](#) I get the difference. the obfuscator piece is what i was missing.

LICENSING:

For my software, I am using a combination of creating a unique number with serial number, mac address number and a few other unique identifiers. This helps me identify a unique machine.

AUTHENTICATION:

Then, i have a web api that authenticates users from the cloud.

PROTECTION(my missing piece):

The piece i was missing was to obfuscate everything so that the code is not crackable.

Remember i am starting and want to protect as much as i can of my intellectual property.

I really appreciate your input and can anyone point me to a post that shows the first 2 (licensing, authentication) items demonstration with a small piece of software? I really want to learn and test my software on my own.

Thanks to all again!

'The piece i was missing was to obfuscate everything so that the code is not crackable.' This is wrong. Obfuscation prevents your source code from being copied or ripped off.

It makes the cracking process harder but it does not prevent it 100%. Obfuscation is a process, and can be reverted back.

Ninja - December 29, 2017, 10:20 am

(Dec 29, 2017 08:51 AM)Martin Heidegger Wrote: [\[CITE\]](#) 'The piece i was missing was to obfuscate everything so that the code is not crackable.' This is wrong. Obfuscation prevents your source code from being copied or ripped off.

It makes the cracking process harder but it does not prevent it 100%. Obfuscation is a process, and can be reverted back.

This.

Move code server sided is the best way to protect it. And if a method on the server has output that changes depending on input then the output cannot be spoofed (ie. a complicated algorithm), so only a user with valid credentials/HWID/IP can use the program correctly.

dotb1n - December 29, 2017, 11:02 am

I am not holding the keys to the kingdom here. I understand security. I was just saying that it definitely makes it harder to reverse each and everything the software does but if anyone wanted my code so bad, and they knew more than I (only as far as the code I have written), then they should have just wrote it themselves.

Ninja - December 30, 2017, 9:21 am

(Dec 29, 2017 12:02 PM)dotb1n Wrote: [\[CITE\]](#) I am not holding the keys to the kingdom here. I understand security. I was just saying that it definitely makes it harder to reverse each and everything the software does but if anyone wanted my code so bad, and they knew more than I (only as far as the code I have written), then they should have just wrote it themselves.

True. If you make your program hard to crack but it is not that 'great of a program' then the process of cracking it is not worth it.

JoshuaSmith - January 4, 2018, 12:05 pm

(Dec 27, 2017 01:29 PM)Ninja Wrote: [\[CITE\]](#) A licensing system is what it says; a licensing system. Most are used with a single line of code. Comment out that line of code, and you 'bypassed' it (with the exception of remote variables and all that). Usually they do not offer anti-crack protection, but only things like encrypting network traffic. To protect your code use an obfuscator to 'scramble' your code and make it hard to read.

getting completely usable source code from anything after .net framework 2.0 without a shit ton of fixing is unreal and i like that people think RE's need readable code. it might make it less readable but not harder.

Ninja - January 4, 2018, 6:00 pm

(Jan 4, 2018 01:05 PM)xmods1337 Wrote: [\[CITE\]](#) getting completely usable source code from anything after .net framework 2.0 without a shit ton of fixing is unreal and i like that people think RE's need readable code. it might make it less readable but not harder.

Of course they don't. Just changing a few opcodes is enough and easy to do. But you can get readable code without issue, then patch the file you want.