What technique can protect a secret from a fully trusted user?

Asked 14 years, 11 months ago Modified 6 years, 8 months ago

Viewed 2k times

7

I am programming a system using C#. My program generates a small message (a hash digest for a file) that I want to store on the hard disk - but I don't want the user to be able to read it. I was going to encrypt this message, but someone has suggested this is A BAD IDEA.

So I'm looking for alternatives - how do you protect a piece of secret information from a fully trusted user?

c# security

Share

Improve this question

Follow

edited Jan 27, 2010 at 23:31

asked Jan 27, 2010 at 22:48

Craig Schwarze

11.6k ● 15 ● 63 ● 81

3 The 'bad idea' suggestion was Eric's comment here: stackoverflow.com/questions/2150703/… – user47589 Jan 27, 2010 at 22:55

6 Answers

Sorted by: Highest score (default)

▲

30

▼

✓

Take a step back; you've got a solution that fundamentally doesn't work for the problem you've got. Instead of trying to hammer on it until it works, stop, step back, and solve the real problem.

Security problems that involve real money are some of the hardest problems to solve; bad people have a real financial motive to attack your system. A multi-pronged approach is usually best for these sorts of things.

First, **write a threat model**:

- identify every resource that needs protecting (your resources AND your benign customer's resources, like their private financial data)
- estimate its value to you
- estimate its value to an attacker
- think of what vulnerabilities expose the resource to attack
- characterize the threat -- who is the attacker and what is their motivation?

Once you know the resources, threats and vulnerabilities, only then start thinking of mitigations to those threats. Assign costs and effectivenesses to each of the mitigations.

For example:

- resource: my TV set
- Value to me: $400
- value to attacker: $40
- vulnerability: unlocked bathroom window
- threat: thieves or vandals use the window to get access to the TV

OK, now that I know what the attacks are, I can start thinking about mitigations:

- lock the window
- get an alarm system
- dogs
- guards

Those are in increasing order of expense. Eventually the cost of the mitigation is larger than the loss of the resource, and it makes no sense to spend the money.

There are also ways to externalize the costs of mitigation:

- threaten the attacker with prosecution -- taxpayers pay for this

- insure the television against theft, reducing the cost of a successful attack against me.
- and so on.

Encrypting a file that contains user data on a user machine is not a mitigation of any attack. Figure out what the attacks are and what *actually* mitigates them, including options like siccing the feds on attackers, and then implement a system that actually mitigates your vulnerabilities and eliminates the threats.

Your proposed mitigation is: give the key to the thief and require the thief to lock the window before he attempts to steal the television. This is not a mitigation of the vulnerability. **No proposal which involves handing the key to the thief is a mitigation of the unlocked window vulnerability, so stop trying to find one.**

For more "software" focused examples of threat modeling, see:

http://download.microsoft.com/download/3/a/7/3a7fa450-1f33-41f7-9e6d-3aa95b5a6aea/MSDNMagazineNovember2006en-us.chm

http://www.owasp.org/index.php/Threat_Risk_Modeling

http://msdn.microsoft.com/en-us/library/aa302419.aspx

And so on; you can find lots of stuff on the web about how we do threat modeling here at Microsoft.

Finally:

Get a security professional involved.

Seriously, you are biting off one of the hardest jobs there is in software implementation, where the consequences of small mistakes have major financial implications. Spend your implementation budget on a top-notch expert consultant who has expertise in this area and can help you find the off-the-shelf and custom-built parts you need to make a secure solution. Rolling your own security system might sound fun and cheap; it is neither. Leave this sort of thing to people who have spent their careers studying this space.

Share Improve this answer Follow

edited Apr 18, 2018 at 13:29
Brian
25.8k ● 18 ● 86 ● 178

answered Jan 27, 2010 at 23:06
Eric Lippert
659k ● 183 ● 1.3k ● 2.1k

7 +1. Your my new favorite person. Your post should be the general go-to post for anyone evaluating security concerns. – Earlz Jan 27, 2010 at 23:24

Thanks Eric, some great thoughts there – Craig Schwarze Jan 27, 2010 at 23:31

7

> My program generates a small message (a hash digest for a file) that I want to store on the hard disk - but I don't want the user to be able to read it.

The user has full control over their machine. If your software can read it, so can the user, with a little bit of effort.

Instead of fighting a losing battle against your customers, it's probably better to accept that "It's the users machine, not mine" and don't bother with anything excessive - just Base64 encode it or something.

Why do you need to stop the user reading a hash digest anyway?

Share Improve this answer Follow

answered Jan 27, 2010 at 22:53

Anon.
59.9k ● 8 ● 84 ● 86

It's an attempt to make a certain log file tamper proof.
– Craig Schwarze Jan 27, 2010 at 22:55

4 The attempt has already failed. You might make it tamper-*evident*, but you won't make it tamper-proof.
– John Saunders Jan 27, 2010 at 23:43

This is the DRM problem, it cannot be done. You can make it very inconvenient and frustrating by coming up

6

with new and novel ways to obfuscate the data, but it is a fundamentally flawed idea to think you can protect data when the machines for encryption and decryption are both hosted in a system the "enemy" fully controls.

Share Improve this answer Follow

edited Jan 27, 2010 at 23:00

answered Jan 27, 2010 at 22:54

Rex M

144k ● 34 ● 291 ● 315

@CraigS: Digital Rights Management en.wikipedia.org/wiki/Digital_rights_management – Austin Salonen Jan 27, 2010 at 23:02

3 @CraigS it means "digital rights management". It's most visible in DVDs and Blu-ray - the attempt to give end-users the ability to only decrypt the data in ways that are "authorized" (e.g. watch it) but prevent copying. It has never worked and it will never work. – Rex M Jan 27, 2010 at 23:02

2

No. It is not possible. If the person has physical/full access to the machine, there is no way you can protect the harddisk on it without encryption.

The only way I could see of doing it is storing this message on a remote server that the user does not have access to.

Share Improve this answer

answered Jan 27, 2010 at 22:54

Follow

Earlz
63.7k ● 100 ● 311 ● 506

Even if he stores it on a remote server when the user runs the program in his machine and the program will access the message the user can get the message. – daniels Jan 27, 2010 at 22:59

Yep... so you could provide them with a server on which to run the application as either a website or ssh with X forwarding. Really though, don't try to hide or obscure things for the computer administrator. It doesn't work and is just annoying. – Earlz Jan 27, 2010 at 23:28

2

Make it so the text can't be of any use without, say, the *other half* of the text which does not evern come close to the users' computer.

In other words: take the value away from the secret, so it won't be a secret any longer nor the customer will have any interest in obtaining it.

If you can't, chances are you did a mistake in designing your app and are trying to find a cheap way to work around it. But there is no free lunch on that. Again, security in this case is yet another example of balancing usability, difficulties of implementation, and value of the secret.

If you still want to do it, don't rely on a single technique. Use many: encrypt the data on disk using a public key that is stored in memory and taken from a repository; do not store the data as 'cleartext' in memory but do block encryption; shuffle memory frequently; obfuscate usage, patterns, latencies, etc.

Take a look at how skype is implemented: through code obfuscation, debugging detection (when a debugger is ran on the executable, the execution path changes), and by taking the actual value of the implementation away. Even if you understand how skype works, it is already a standard; and if you want to reverse-engineer it to get your software to "work with skype", well, they'll never give you auth to use their brand.

Share Improve this answer Follow

answered Jan 27, 2010 at 23:04

lorenzog

3,611 ● 4 ● 32 ● 51

2

In one of the comments to another answer, you wrote:

> It's an attempt to make a certain log file tamper proof.

If that's really the case, then perhaps you should take a different approach and move the log file to a place where the user is *not* fully trusted (ie: some other system).

Otherwise, you cannot guarantee the security of the file (as others have mentioned).

Share Improve this answer

answered Jan 27, 2010 at 23:38

Follow

Craig Walker
51.6k ● 57 ● 155 ● 212