How do you protect code from leaking outside? [duplicate]

Asked 16 years, 2 months ago   Modified 9 years, 6 months ago

Viewed 5k times

Besides open-sourcing your project and legislation, are there ways to prevent, or at least minimize the damages of code leaking outside your company/group?

We obviously can't block Internet access (to prevent emailing the code) because programmer's need their references. We also can't block peripheral devices (USB, Firewire, etc.)

The code matters most when it has some proprietary algorithms and in-house developed knowledge (as opposed to regular routine code to draw GUIs, connect to databases, etc.), but some applications (like accounting software and CRMs) are just that: complex collections of routine code that are simple to develop in principle, but will take years to write from scratch. This is where leaked code will come in handy to competitors.

As far as I see it, preventing leakage relies almost entirely on human process. What do you think? What precautions and measures are you taking? And has code leakage affected you before?

`security` `project-management` `source-code-protection`

Share

Improve this question

Follow

---

3   How does open sourcing your product prevent code leaks? Surely thats like saying "waving a white flag and sayying we surrender is a good way to prevent war"... – Omar Kooheji Oct 6, 2008 at 12:08

It all boils down to least privilege and lowering the blast radius, this sums it up pretty nicely: blog.arnica.io/…
– Eran Medan Oct 7, 2021 at 17:32

---

14 Answers

Sorted by: Highest score (default) ⇕

▲

43

You can't stop it getting out. So two solutions - stop people wanting to hurt you, and have legal precautions. To stop people hating you treat them right (saying more is probably off topic for stack overflow).

I'm not a lawyer, but to give yourself legal protection, if you believe in it, patent the ideas, put a copyright notice in the code, and make sure the contracts for your programmers specify carefully intellectual property rights.
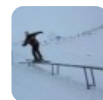
But at the end of the day, the answer is run quicker than the competition.

Share Improve this answer

Follow

1   Eloquently and to the point, nice. – Torbjörn Gyllebring Oct 6, 2008 at 11:15

1   If you make them feel like they are part of the team and that they would be hurting their friends by leaking code that would be a start... – Omar Kooheji Oct 6, 2008 at 11:19

4   "Treat them right [i.e. your developers]". Wholeheartedly agree. – mdec Oct 6, 2008 at 11:30

Completely right - but consult with a lawyer, and if the lawyer doesn't say at least that you need a policy that software never leaves the building then find another lawyer. – plinth Oct 6, 2008 at 12:58

17

Unless you're working with something highly classified and given that you can't block email and USB devices I guess you aren't there's really not to much damage to be had even if the source code leaks. The thing is, what is

the code, or parts of it worth without the knowledge of how it works and the organization around it.

In general the value of "source" is much less than is commonly touted, basicly the source without the people or the organization isn't worth the storage it occupies for a competitor.

Also, you're missing the most likely attack vector, and it's also the one you can't stop no matter what. If someone really really want's to know how you made your magic then they'll try to hire your developers away, and since you can't stop them from having information inside their skull and even if they turn in all their possesions ther knowledge and domain expertise is leaving with them. Basicly employee retention and trust is the only way. Sorry.

Share   Improve this answer

Follow

answered Oct 6, 2008 at 11:14

Torbjörn Gyllebring
18.2k ● 2 ● 31 ● 22

> Yes, the competitor will often gain more if he just hires your developer, even not getting the code, than if he gets whole product source code and have no knowledge and experience with it. – Jacek Konieczny Mar 23, 2010 at 11:32

I don't know how much actual help this is going to be, but:

13

1. Don't p*ss your programmers off. Don't get them in a position where they *want* to give the source to a

competitor. Most places undervalue their developers. Given where you are (SO), I guess you are less likely to. Nothing got to me more than seeing the sales folks out for games of golf - paid, and paid for, by the company - while we had to fight to get pizza once a month.

2. Really, if your direct competitors got your code *today*, what would it do? Is your product or vertical market that stagnant that you wouldn't release newer, better versions before they could react? Is there no room for innovation? Most companies overvalue their "proprietary algorithms and in-house developed knowledge". Sure, it may cut some time off, but it's only about 10% of the problem.

3. If you got all the source for all your competitors products, how much actual use would it be? I'd guess it would set you back months. Not forward. Back.

If you had a clean system, and little external/internal knowledge, how long would it take you to get your own product into a buildable state? How long would it take to drill down into the code and workout what is going on? How much time and money would you waste trying to work something out, rather than spending time and money on how to make *your* product work better?

I've actually been in the position of having all the source - 1million lines+ of code - to a competitor's product. We did nothing with it - aside from a bit of a poke-around and

then delete it, which was more than I was comfortable with - but I would expect that we'd have chewed up months of time just to get to where they were then.

So we nuked it, slapped the id10t who got it (yes, a developer/PM who came over from the other company), and thought about how to make *our* product kick so much butt that it didn't matter what they did. Much better use of time. Worked well, too. We had differentiators, not just re-hashing the same features in the same way they did them.

Sorry, but there is *no* way you can stop people getting stuff out, and still be able to actually work. You can stop them *wanting* to do it, or make it so there is no value to them having it.

We were worried about people decompiling our code too. We stopped worrying when we realised that WE had enough trouble working out what was going on inside 500K+ lines of C#, C++ and HTML code talking to MAPI/Exchange. If someone can decompile it and work it out, then we want to hire them......

BTW, for clarity, and given who I now work for, I should point out this is *not* my current employer. This was quite a while ago.

Share   Improve this answer

Follow

edited Apr 3, 2014 at 20:00

halfer
20.4k ● 19 ● 108 ● 200

3   +1 for `Dont p*ss your programmers off.` — user Apr 10, 2012 at 4:23

▲

8

▼

🔖

🕔

The code does not leak out on itself. It takes people to take it. There are obviously some security measures you might use like traffic analysis and lock-down on the repositories so only authorized developers can connect to it.

But by the end of the day your best option is to make sure that no one WANTS to steal from you. Your team has to be happy, they have to be proud to work for your they have to be loyal to the company and to each other. If you have such team it's a simple question of explaining to everyone that the code has to be protected from outsiders. It will not stop a dedicated mole but will prevent accidents.

P.S. And yes, proper clauses in the contracts would not harm as well, at least they will make sure that the developers are AWARE that taking code outside is morally wrong.

Share Improve this answer

Follow

Okay, I am going to be a little practical here.

- Being nice to everybody and hoping they won't hurt you doesn't work.

Every programmer knows from the day he joins a company that he'll not stay there forever. He will change when he's learned enough to get a better opportunity.

The programmers who write the code believe that they have the ownership to it even if they wrote it on the time they rented out to somebody else. So many of them will usually try to get their hands on the source-code even if they don't intend to hurt anybody.

Once they leave the company and they've carried the source code with them and lost contact with their colleagues, the conscience settles down and goes on a vacation and after a while bits and pieces from the code start showing up everywhere.

That's what I KNOW happens cause I've witnessed it happen to my company.

So what does one do?

- Sign a NDA which specifically mentions that they programmer WILL not take copies.

- Distribute your product between programmers, and if possible get modules coded individually and

integrated by a chief whose responsibility is that all programmers do nt get all the code.

- At the time of termination get a written undertaking from the coders that they do not possess any IP of the company and they understand the penalties of violation.

- If somebody violates your IP, sue the man! No exceptions. It'll work as an example for the present team.

Do I sound extreme?

Share   Improve this answer

Follow

answered Jan 6, 2009 at 16:45

Cyril Gupta
13.7k ● 11 ● 69 ● 91

Not at all!!!!! – busaud Apr 20 at 16:49

▲

6

▼

Follow these guidelines and it shouldn't matter if the contents of your entire source code repository is posted all over stackoverflow:

http://geocities.com/mdetting/unmaintainable.html

Oh, and show your developers that you don't trust them by blocking access to parts of the source code, scanning outgoing/incoming email etc. That is a surefire way to make them want to stay around... ...nothing improves morale like a bit of mistrust in the workplace.

Another cool way is to tell one half that they are "team a" and name the other half as the untrustworthy "team b". Then reverse it and say the same thing to the "team b" members. Encourage them to keep an eye on the "bad guys" in the other team and to report any signs of illoyalty to you. Sprinkle a few "conflict inducers" (e.g. tell "Joe": '*do you know what Ed says about you behind your back?*') etc. Works wonders if you set up the developers against each other and create a few [invented-by-you] conflicts here and there...

(Eh, and no, I don't actually recommend any of the above. Just kidding. But I have seen people use all of the tactics above. And it didn't work.)

Share Improve this answer

Follow

answered Oct 6, 2008 at 11:12

KristoferA
12.4k ● 1 ● 43 ● 63

I remember this happening to Valve when they were developing HL-2. Interesting link here: http://www.shacknews.com/onearticle.x/28619

Share Improve this answer

Follow

answered Oct 6, 2008 at 11:49

endian
4,294 ●8 ●37 ●44

Most of the answers are based on Moral and ethical values. I wonder if Google, Facebook etc. just rely on their employees good will. Give me a break, that's totally utopian. Don't be a fool. Be realistic.

YES, it is possible to prevent code leaking:

Using a virtual server hosting virtual machines, programmers can only access locally to these virtual machines (intranet) via Remote Desktop. Repository is managed locally. private keys are required to access the repository. **Copy/paste from virtual machine to client is disabled. only copy/paste from client to virtual is allowed.**

Companies like facebook do that.

The only way to still code is by taking pictures to the actual code, which is totally not practical and feasible at all, and since there are surveillance cameras everywhere, you will have to go to the bathroom to take those pictures.

1

I've worked somewhere where there was a real culture of secrecy about this sort of thing (historically there had been a number of times when the company was small where "customers" had, shall we say, abused their access to our product).

While at the top the management were very protective, I see it slightly differently. I think our code, while not entirely irrelevant, isn't as key as you'd expect it to be in a software company.

The reason that we are successful is:

1) The code is essentially the solution to a bunch of problems. If you get our code you get those solutions but we still have the smart people who solved those problems. They understand those problems better than you do and are better able to solve the next set of problems better than you are.

2) Because they really understand the problems (and the solutions) we can do things faster than our competitors which translates to cheaper (or more profitable).

3) Also because of those people and the attitude within the company we've delivered well to our clients and provided good support.

4) And because of that we have a good reputation and reference-able customers.

A small number of companies have code which is genuinely worth keeping secret - proprietary algorithms and that sort of thing - but for a vast majority of us our products are very easily replicable by smart people.

What I'm saying is do the basics - write it into people's contracts that they can't take it, keep it secure and so on - but don't obsess over it. Unless you're in a very specific market it's unlikely to be what's really going to make your business succeed or fail.

Share Improve this answer

Follow

answered Mar 23, 2010 at 11:15

Jon Hopkins

2,263 ● 25 ● 28

0

The best step starts from reruting guys with strong ethical behaviour. Various other steps can be taken like all communication being scanned. There are places where email and all information going out is scanned. The desktop/laptop does not have hard-disk or the access is restricted and all work is on network folders, even when working from home, one has to get connected to internet. The offline work gets synchronized. The USB and drives are disconnected.

The other policies are to provide access only on need basis. These will only slow down and hinder to some extent, but is one is very determined then he would find

ways to get around this. The other way is if the code is really very important, then have the idea copywrite protected legaly.

Share   Improve this answer

Follow

Dheer
4,066 ● 6 ● 37 ● 45

> You can't rely on your instinct and opinion to determine whether new recruits are ethical or not. Plus, if anything goes wrong, you'll hear a lot of "But *you* hired him." – rami Oct 6, 2008 at 11:18

> I've never had any ethical questions on an interview, and the only times I've been told things like "don't publish live customer data" other than an ad hoc, spur of the moment thing, was when I was working in a regulated industry. – David Thornley Jan 6, 2009 at 16:38

0

To be honest it's almost impossible. If I wanted to suggest what a company that would shortly appear on the Daily WTF would do:

1. Disconnect the "work computer" from the internet, bt because they need internet access for reference buy everyone a wbbook.

2. Stuff the developers USB slots with epoxy and require that they load/unload everything from a centralised server, which scans all the data that goes through it for code like syntax.

Or you could just trust your employees and make them sign an NDA...

Share   Improve this answer

Follow

answered Oct 6, 2008 at 11:18

Omar Kooheji
55.6k ● 71 ● 186 ● 243

I personally never tested on any real case, but I would suggest using code fragmentation:

basically you split your project in a number of libraries, define interfaces and unit tests for each of them, then you separate SVN repositories so that each group have access to a limited part of your precious source code.

This is also a good practice no matter what and should help if you are outsourcing abroad.

Share   Improve this answer

Follow

answered Oct 6, 2008 at 11:20

Pokot0
562 ● 1 ● 7 ● 12

1   ...and it really makes troubleshooting/debugging near impossible... – KristoferA Oct 6, 2008 at 11:27

Yeah, it is a sure way to antagonize the employees. Put up too many hoops and people will refuse jumping through at some point. – HS. Oct 22, 2008 at 12:25

Make it very clear that you don't trust your people, and there's this chance they'll pick up on that and become untrustworthy. People do tend to live up or down to expectations. – David Thornley Jan 6, 2009 at 16:36

The previous answers all seem to center on building trust and employing ethical people.

Another possibility might be to create your own domain specific language and tools. That will make any leaked code harder to use. It might still be possible to steal useful ideas from it, but it would not be possible to simply compile a competing product unless the whole toolchain is leaked.

Share   Improve this answer

Follow

answered Oct 6, 2008 at 12:05

HS.
2,615 ● 2 ● 21 ● 29

That looks like a way to get to the DailyWTF. Doing your own language/framework is a bad idea in most cases, even if decided for technical reasons. Building your own tools just because of fear of the competition is like giving the competition a favour. They will just have an easier job to getting their product done before yours is ready.
– Jacek Konieczny Mar 23, 2010 at 11:40

Good point, it might not be easy. A good API needs expert knowledge, but own languages can also simplify development due to being tailor-made exactly for the intended purpose. That is another reason for "domain specific language and tools", they can make your work even easier.
– HS. Mar 24, 2010 at 4:01

Trust your developers. People tend to live up or down to expectations. Treat them well, and remember that loyalty goes both ways. After all, if you can't cut off thumb drives, you can't stop anybody from leaking code, no matter how much you don't trust them.

That being said, find yourself a lawyer with trade secret expertise, probably expertise in other parts of IP law, and ask how to legally safeguard stuff. You do want to make sure that, if a competitor gets your stuff, it's not legal for the competitor to benefit from it.

Share   Improve this answer

Follow

answered Jan 6, 2009 at 16:44

David Thornley
57k ● 9 ● 95 ● 158