What are the best practices to keep your ethereum smart contract code from being copied and reused?

I'm new to smart contract development. I have built a couple of sample projects and deployed those to my private network. So now I am doing some research on best practices in regards to smart contract security and protecting my intellectual property from those that would like to steal it and use it for their own purposes.

From what I have read, smart contract bytecodes are visible in the blockchain, but supposedly it is impossible to turn those bytecodes back into solidity code. Even still, is there anyway to take the bytecodes from the block chain, copy them and use them for your own purposes?

Please give me some best practices to follow when dealing with smart contracts in regards to security and protecting my intellectual property.

solidity truffle dapp-development security

asked Aug 14 at 21:18

Jason Cochran
62 11

2 Answers

Sorry to be a downer...

You can't recover the original code from byte code because information is lost during the compilation. However, disassembly is **always** possible if the byte code is retrievable. Since the code for the contract must be executed by all full nodes and by miners, the byte code must be readily available to those parties. In practice, since anyone can be a node, that means everyone can get the byte code. As a former assembly programmer, I will assert that it is impossible to protect your IP completely in this case. If it's economically beneficial or just plain interesting, it will probably get reverse engineered.

Unfortunately, I can't offer best practices; legal avenues may be your best bet (but not patenting or copyright; not particularly useful here). At best, I think you can use some anti-disassembler techniques, but those will increase the cost of executing your contracts. As a person who has personally stripped out anti-piracy code from software I own in order to speed it up, I would ask that you try some other method of making your business (I assume) more resilient to reverse engineering.

Given that even with protections like clock-skew detection, instruction timing checks, hardware-based protection, encrypted instructions, and checks that test physical hardware only slow down the best crackers, it's almost certainly a better use of your time/efforts to build a better product/business than it is to try to defend against this scenario on the EVM (which is a much simpler system than a real computer and therefore easier to defeat due to a reduction in things that can be worked into the byte code). If you have to ask this question about best practices, it's almost certain that there are people, individually (let alone as a group) who can break your obfuscation efforts more easily than you can implement obfuscation.

edited Aug 15 at 2:44

answered Aug 15 at 2:09

lungj
2,723 1 4 27

OK let's say that I am an entrepreneur who has $10m VC funding to build something on the Ethereum block chain. I hire a large team of developers and build some awesome dapp and deploy my smart contracts. A month down the road they discover that some would be "hacker" copied the deployed contract bytecodes and is using them for their purposes; ie: 100% ripping off the original development. So in the above situation, you are saying the is no way to prevent this from happening? To me that seems really bizarre. Who in their right mind would want to use Ethereum to create a product? – Jason Cochran Aug 15 at 19:56

Tongue-in-cheekily (another answer to follow as a comment): Let's say it's 1981 and you're IBM and have a product called a "personal computer" and have partnered with a small company named Microsoft to market and sell a product called MS-DOS. Later, they discover that some would-be "hackers" can easily copy the software and sell it for themselves. Who in their right mind would want to create a product for an IBM PC? – lungj Aug 15 at 20:05

1 As I'm sure any VC will tell you, they're usually not investing in products, they're investing in people. They're investing in the team's ability to execute. For example, several open source projects make money without selling their software. Pepsi has a "Coca-Cola clone" and I'm sure many other companies have tried to dethrone this duopoly. The patent for acetylsalicylic acid, has expired; why do people keep buying the brand name Aspirin instead of a generic which has, ostensibly, the same product? – lungj Aug 15 at 20:10

Fair enough... you think I'm overreacting? Granted I'm ignorant on many aspects of Ethereum. If a client hired you to build a dapp, would you recommend using a private node? – Jason Cochran Aug 15 at 20:13

3 No, I think your reaction is perfectly justified. However, you might want to rethink your business model. There's a lot more to companies than their technologies. There are things like trust, support, first-mover, marketing, vendor lock-in, and more. You can use these to your advantage! Instead of seeing this transparency as something that will hobble you, perhaps it's better to think of it as a strength: instant trust that you're not hiding a ticking time bomb in your code. By asking, you're understanding what you're getting into which is probably 10 steps ahead of your competitors! – lungj Aug 15 at 20:15

I sometimes consult to people about smart contract development, deployment, security, etc. I tell all of my clients that they should never interact with a smart contract if it's not fully open source.

We do not want to train people to interact with code that, if it's not open source, can do anything it want with your money.

Your business model should include fully open solidity source code. If it does not, I think (and I hope) you get very few adopters.

answered Aug 15 at 5:02

Thomas Jay Rush
4,031 8 37