TRUSTLOOK

Understanding Trustlook Smart Contract Guardian

Smart Contract Guardian

Smart Contract Guardian is an online tool to decompile EVM bytecode into more human readable format, and it also inspects the bytecode for common vulnerabilities of smart contracts.
You can specify your EVM bytecode by 2 ways: Address or Copy&Paste. If you have any concern about the inspection result of the smart contract, please feel free to contact us for a manual audit.

Solidity Decompilation
Retrieve Solidity like code from bytes Recognize multiple storage types

Function names
Retrieve original function names

Stability
Tested on millions of real world contracts

Select "Paste hex" or "Smart contract address" then paste the hex string of the EVM binary or the address, then click "Decompile Now"

DECOMPILE NOW REQUEST AUDIT

https://www.trustlook.com

Introduction

“Ethereum is a decentralized platform that runs smart contracts” - from Ethereum Foundation. Since Ethereum implemented the first block chain network with smart contracts, which have brought many followers who use Ethereum Virtual Machine (EVM) compatible bytecode in their smart contracts too. One ironic thing is that for a decentralized platform it is claimed transparent and open to all participants. However, not all of the smart contract authors published their source code online. In fact from our testing, there are more than 80% smart contracts running on main network of Ethereum have no source code available. In order to audit these binary smart contract, you need to be very familiar with the meaning of the Ethereum Virtual Machine (EVM) bytecode. Without high level source code, the audit on these EVM binary can be very difficult and time consuming. However, the audits can’t be passed since any of the vulnerability found in these contracts will result in real financial lose.

Trustlook

Since 2017, Trustlook has noticed the security concerns in block chain industry, especially for the EVM bytecode level audit. There was no really working solution to help these binary code auditors and researchers. To understand the logic of a binary EVM smart contract can be extremely hard and tedious. Even for the contracts who claimed to have published their source code, you still need to verify whether the binary EVM codes was originally compiled from the claimed code. The recent TRX stolen event just shows a good example when

compiled code was not verified. To help the situation in block chain community, Trustlook has published its Smart Contract Guardian solution for free. By using the decompiler tool, your EVM bytecode audit will be much easier.

Solidity Decompilation

Trustlook Smart Contract Guardian solution retrieves Solidity like code from EVM byte code, which will prevent you reading the low level machine code and being lost in the tedious control flow graph. It will recognize multiple storage types like structure, multiple level mappings, which extremely simplified the binary audit process.

Retrieve original function names

In Evm platform, the original function names were replaced with a 4-byte hash value. So theoretically, it is not possible to retrieve the original function names back from EVM byte code. However, by calculating all function names in existing source code, we can build a huge function mapping table. It will help our solution to retrieve back more than %99 the original function names back.

Stability

To make our solution stable and practical to use, we have tested millions of smart contracts running in the real network. Considering the fact that these EVM byte code was compiled by all kinds of EVM compilers and different versions, our Smart Contract Guardian can always output readable Solidity like code at the end.

Free to use

To contribute to the blockchain community, we decided to publish the tool for free. Any EVM auditor or researcher can use it without any toll. Together, we will make the smart contracts more safe and efficient to run in the network.

Easy to use

The Smart Contract Guardian portal is easy to use. By either enter the EVM byte code or a valid Ethereum smart contract address, you can click on the "*DECOMPILE NOW*" button to see the result.

```
163     mapping(address => mapping(address => uint256)) public allowance
164
165     function transferOwnership( address arg0) public return ()
166     {
167         require((uint160(msg.sender) == uint160(uint160(owner))));
168         require(!(uint160(arg0) == uint160(0x0)));
169         log(0x80,0x0,0x8BE0079C531659141344CD1FD0A4F28419497F9722A3DAAFE3B4186F6B6457E0,uint160(uint160(owner
)),uint160(arg0));
170         owner = uint160(arg0);
171         return();
172     }
173
174     function func_00000DC1( uint256 arg0,uint256 arg1) private return (var0)
175     {
176         allowance[uint160(uint160(msg.sender))][uint160(uint160(arg0))] = arg1;
177         mstore(0x80,arg1);
178         log(0x80,0x20,0x8C5BE1E5EBEC7D5BD14F71427D1E84F3DD0314C0F7B2291E5B200AC8C7C3B925,uint160(msg.sender
),uint160(arg0));
```

Summary

The founding team at Trustlook are cybersecurity veterans with over ten years of industry experience, with deep understanding of traditional cybersecurity fields as well as cutting edge blockchain security issues. Trustlook seeks to provide security and reliability to all smart contract based services in order to build a safer and more mature Ethereum compatible networks.