Transper: State Extraction and monitoring of Ethereum Contracts

We are building an Ethereum tool, Transper that applies static/dynamic analysis techniques on a deployed smart contract in order to extract all variable values from its storage state including complex structures like arrays, mappings. Visibility into the actual values of smart contract variables is helpful in improving code comprehension, developer debugging, testing of contracts. Currently, regular variables in solidity can be easily obtained, but complex structures like mapping require intelligent analysis of storage slots. This is because, complex map variables, the key-value pair is stored randomly in the Ethereum storage. Our tool builds a source code-driven algorithm for safe analysis and extraction of index keys of map structure specifically and identifies cases when all variables can be safely extracted. In case, it cannot extract a value(e.g. index key to mapping is not known), it safely reports it. Our initial results show that for 90% of the time, the state of smart contracts with mapping variables can be extracted safely. Further, we have successfully extracted a snapshot of the state of several smart contracts and redeployed a newer version of the smart contract with the snapshot state reinstated.

In one case study, the current Transper tool was able to complete the analysis for the key origin in the mapping structure of 643 deployed contracts, thereby able to extract all state of the smart contract. It could handle various versions of Solidity compiler, multiple parent contracts, inheritance hierarchies. There were 3696 functions in the 643 contracts in which a mapping variable declared inside the contract was being modified and there were 1128 mapping variables in total in those 643 contracts. Transper's static/dynamic analysis identified a total of 4969 keys used to modify the mappings, coming from the following origins :

- Arguments of Functions (4727)
- Static Values(137)
- State Variables(75),
- Runtime Variables (30).

Here Arguments of Functions refers to the arguments passed to the function in which a mapping variable was modified, the static values refer to hardcoded values in the smart contract code, state variables are the global variables which are stored on the storage trie and are accessible throughout the contract and Runtime variables are variables which are created when a function is called and their scope is limited to the function in which they were generated i.e variable created to iterate a loop.

The prototype of the tool can be found on the following link.

https://github.com/blockchain-unit/Transper-CLIt

Please report the issues on the repository if you face any.

The tool works with multiple inheritance and for all data types but very complex data types like mapping of mapping and multidimensional static + dynamic arrays require additional techniques to be implemented.

For now, we are extracting the keys of the maps from the arguments of the functions and hard coded values but we can further extend our technique to look for values of keys in the logs of Smart Contract as many contracts emit events to store values in the logs.