Advanced Topics in Software Engineering
CSE 6324 - Section 001

INCEPTION

Team 4

Suvarna, Arjun - 1002024437
Sinari, Navina - 1002072310
Palnati, Netra - 1002030626
Waje, Sanjana - 1002069940

Background

- The ecrecover() function is vulnerable to replay attacks if not used without proper checks, like adding nonces. (Github Issue #1950) [6] [9].
- Unencrypted data like keys and passwords on chain stored even in private variables are visible to a attacker [7].
- Hardcoded credentials make it easy for an attacker to use them for nefarious purposes [8].

Inception

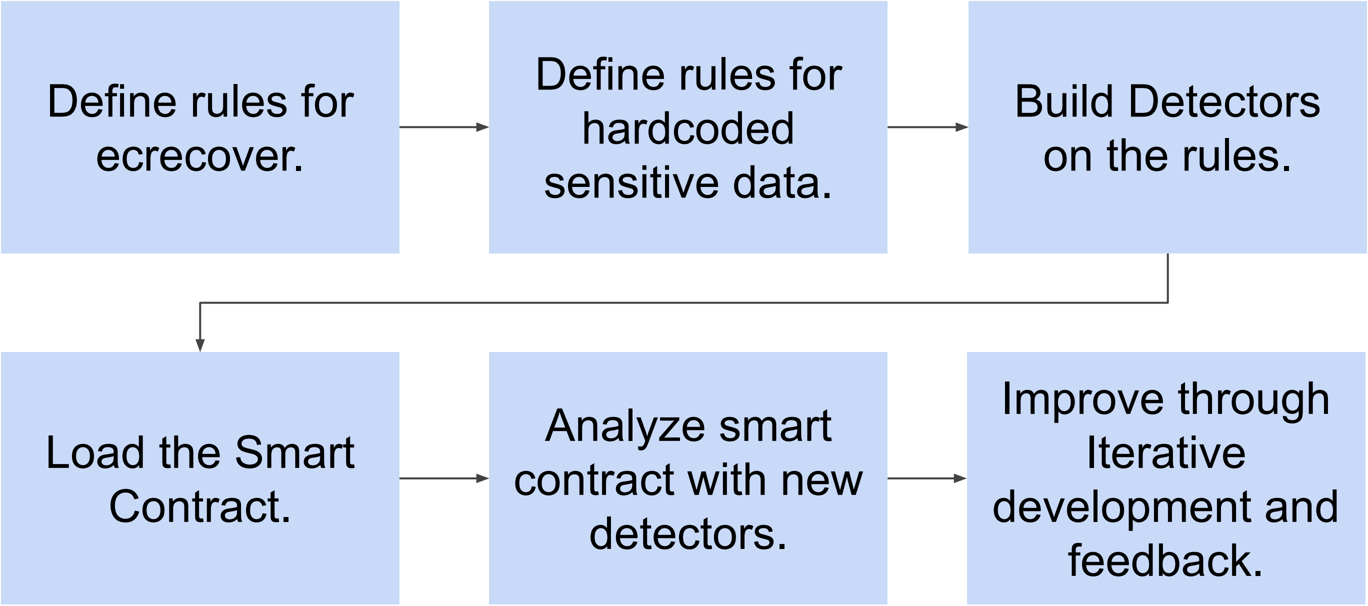
New detectors to detect:

1. Usage of the ecrecover() function without nonce and flag it as a security vulnerability.
2. API Keys stored as plain text.
3. Hard-coded passwords.

Risks

1. Incomplete rule definition
2. False Positives
3. False Negatives
4. Change in Solidity standards

Planning



Define rules for ecrecover.

Define rules for hardcoded sensitive data.

Build Detectors on the rules.

Load the Smart Contract.

Analyze smart contract with new detectors.

Improve through Iterative development and feedback.

Competitors

1. **Slither** : Slither is a static code analysis tool [11]. Slither currently does not have a public detector for API Keys, Passwords, or a public detector for improper usage of the ecrecover function [9].

2. **Mythril** : Mythril is a security analysis tool for Ethereum smart contracts [2]. It currently has no modules to detect unprotected use of the ecrecover function or hardcoded credentials [12].

Competitors

3. **MythX**: MythX is a tool that scans for security vulnerabilities in Ethereum and other EVM-based blockchain smart contracts [13]. It does not have a detector available for improper use of ecrecover or hardcoded API keys and passwords [3].

Customers and Users

1. **Smart contract developers** with explicit signature checks: Ethereum smart contracts generally use the ecrecover() function to verify. Here, users can elect to explicitly verify the signatures to save on gas [4].

2. **Wallet Providers** that use ecrecover to verify signatures: Applications for wallets like MetaMask and Toshi assist with signing transactions that are verified using ecrecover() [5].

Customers and Users

3. **Smart contract developers** with authenticated external calls:
   Developers who use external calls through an oracle with an authentication mechanism for the external calls [10].

Github Repository

https://github.com/arjunsuvarna1/CSE6324_Team4_Fall23

References

[1] Unencrypted Private Data On-Chain- https://swcregistry.io/docs/SWC-136/

[2] Mythril- https://github.com/Consensys/mythril

[3] MythX- https://mythx.io/about/

[4] Ecrecover function in Solidity - https://www.educative.io/answers/what-is-an-ecrecover-function-in-solidity

[5] Multi-signatures for Ethereum - https://medium.com/dsys/now-open-source-friendly-multi-signatures-for-ethereum-d75ca5a0dc5c

[6] SWC-121- https://swcregistry.io/docs/SWC-121/

[7] SWC-136 - https://swcregistry.io/docs/SWC-136/

[8] CWE-798 - https://cwe.mitre.org/data/definitions/798.html

[9] Improper usage of ecrecover - https://github.com/crytic/slither/issues/1950

[10] Make HTTP Request Using Your Solidity Smart Contract - https://medium.com/coinmonks/make-http-request-using-your-solidity-smart-contract-4f7173bd391c

[11] Slither Detector Documentation - https://github.com/crytic/slither/wiki/Detector-Documentation

[12] Mythril Analysis Modules - https://mythril-classic.readthedocs.io/en/develop/module-list.html

[13] Multi-signatures for Ethereum