Automatically Fixing Vulnerabilities in WebAssembly

Yubin Hu

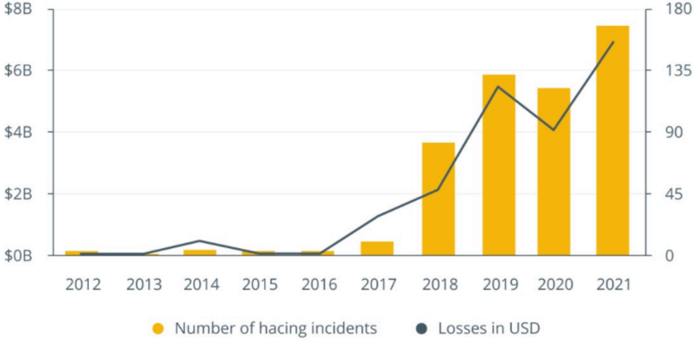
yubin.hu@bupt.edu.cn

December 7, 2021

Agenda

- **Backgroud: the importance of contract security**
- Vulnerabilities in smart contracts
  - Reentrancy
  - Missing Input Validation
  - Unhandled Exception
  - Arithmetic Vulnerabilities
  - Fake EOS
  - Fake Receipt
  - Rollback
  - Missing Permission Check
- Vulnerabilities Detection
- Automatic Fixes
- Evaluation
- Reference

Backgroud: the importance of contract security



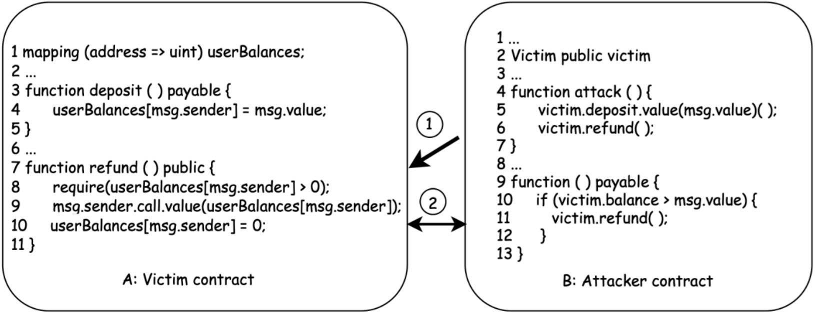
Agenda

- Backgroud: the importance of contract security
- Vulnerabilities in smart contracts
    - **Reentrancy**
    - Missing Input Validation
    - Unhandled Exception
    - Arithmetic Vulnerabilities
    - Fake EOS
    - Fake Receipt
    - Rollback
    - Missing Permission Check
- Vulnerabilities Detection
- Automatic Fixes
- Evaluation
- Reference

Reentrancy

- init
    - Attach account: $100
    - Employee A
    - Employee B
- attack
    1. Request a withdrawal of $60 from Employee A.
    2. Employee A gives $60 to the attacker
    3. Request a withdrawal of $60 from employee B. At this time, employee B does not know that the attacker has already withdrawn $60 from employee A.
    4. Employee gives $60 to the attacker.
    5. Employee B changes the balance of the attacker's bank account.
       Attacker's account:100 - 60 = 40
    6. Employee A changes the balance of the attacker's bank account.
       Attacker's account:40 - 60 = -20

Reentrancy



```
mapping (address => uint) userBalances;
...
function deposit ( ) payable {
    userBalances[msg.sender] = msg.value;
}
...
function refund ( ) public {
    require(userBalances[msg.sender] > 0);
    msg.sender.call.value(userBalances[msg.sender]);
    userBalances[msg.sender] = 0;
}
```

```
...
Victim public victim
...
function attack ( ) {
    victim.deposit.value(msg.value)( );
    victim.refund( );
}
...
function ( ) payable {
   if (victim.balance > msg.value) {
      victim.refund( );
   }
}
```

Figure: An exploit of Reentrancy vulnerability

Agenda

- Backgroud: the importance of contract security
- Vulnerabilities in smart contracts
  - Reentrancy
  - Missing Input Validation
  - Unhandled Exception
  - Arithmetic Vulnerabilities
  - Fake EOS
  - Fake Receipt
  - Rollback
  - Missing Permission Check
- **Vulnerabilities Detection**
- Automatic Fixes
- Evaluation
- Reference

Vulnerabilities Detection

- Difference comparison
- Find control flow, data flow characteristics
- Fuzzing
- Using multiple existing tools, and seting thresholds to determine if it is a vulnerability

Agenda

- Backgroud: the importance of contract security
- Vulnerabilities in smart contracts
  - Reentrancy
  - Missing Input Validation
  - Unhandled Exception
  - Arithmetic Vulnerabilities
  - Fake EOS
  - Fake Receipt
  - Rollback
  - Missing Permission Check
- Vulnerabilities Detection
- **Automatic Fixes**
- Evaluation
- Reference

Automatic Fixes

- Generates patches using template-based fix patterns and leverages static program analysis
- Binary rewriting. Binary rewriting has also been applied to retrofit security hardening techniques such as control-flow integrity, to compiled binaries, but also to dynamically apply security patches to running programs. For binary rewriting on traditional architectures two flavors of approaches have been developed: static and dynamic rewriting.

Agenda

- Backgroud: the importance of contract security
- Vulnerabilities in smart contracts
  - Reentrancy
  - Missing Input Validation
  - Unhandled Exception
  - Arithmetic Vulnerabilities
  - Fake EOS
  - Fake Receipt
  - Rollback
  - Missing Permission Check
- Vulnerabilities Detection
- Automatic Fixes
- **Evaluation**
- Reference

Evaluation

- False Positives
- Runtime Performance
- Extra Gas

Agenda

- Backgroud: the importance of contract security
- Vulnerabilities in smart contracts
  - Reentrancy
  - Missing Input Validation
  - Unhandled Exception
  - Arithmetic Vulnerabilities
  - Fake EOS
  - Fake Receipt
  - Rollback
  - Missing Permission Check
- Vulnerabilities Detection
- Automatic Fixes
- Evaluation
- **Reference**

Reference

[1] Q. Le, "How hackers attack eos contracts and ways to prevent it," nov 2018.

[2] N. He, R. Zhang, H. Wang, L. Wu, X. Luo, Y. Guo, T. Yu, and X. Jiang, "EOSAFE: Security analysis of EOSIO smart contracts," in 30th USENIX Security Symposium (USENIX Security 21), USENIX Association, aug 2021.

[3] D. Wang, B. Jiang, and W. K. Chan, "Wana: Symbolic ex- ecution of wasm bytecode for cross-platform smart contract vulnerability detection," ArXiv, vol. abs/2007.15510, 2020.

[4] "Eosio developer portal."

[5] "Webassembly 1.0 has shipped in 4 major browser engines.."

Reference

[6] ““transactions protocol,”.”

[7] “”fake eos attack” upgraded, 60k eos tokens lost by eoscast.” [8] “Roll back attack about blacklist in eos.”

[9] ““dappradar, a dapp browser,”,” dct 2020.

[10] “Dapptotal,,” nov 2019.

[11] ““blogs about blockchain security events,”,” 2020.

[12] ““blockchain security events,”.”