Visualizing Security

Raine Rupert Revere
github raineorshine
twitter @metaraine

How do you spot smart contract security vulnerabilities?

You know, to prevent people
from stealing millions of dollars.

Common Attacks

➔ **Array Griefing**

```
for(uint i=0; i<arr.length;
i++) { ... }
```

➔ **Reentrancy**

```
address.value(balance)();
```

➔ **Underflow**

```
balance -= amount;
```

What do each of these attacks have in common?

They all have specific code smells.

If we can detect these code smells, we can help prevent these errors.

Static Analysis

Static analysis is a method of testing and evaluating a program without executing its code.

solidity-parser

github.com/consensys/solidity-parser

- ➔ In: Contract Source Code
- ➔ Out: Abstract Syntax Tree (AST)

solidity-parser

github.com/consensys/solidity-parser

- ➔ **In:** **Contract Source Code**
- ➔ **Out:** **Abstract Syntax Tree (AST)**

An Abstract Syntax Tree is like a “map” of your code that can be traversed and explored programmatically.

solgraph

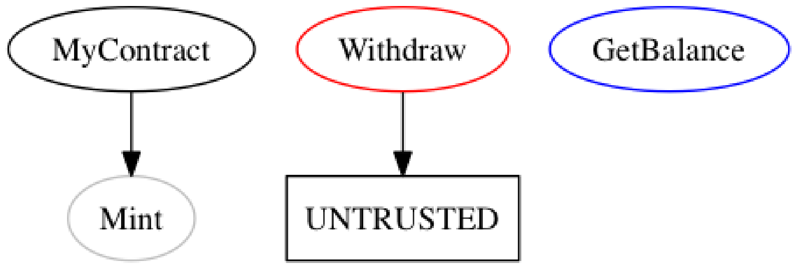
github.com/raineorshine/solgraph

- ➔ In: Abstract Syntax Tree (AST)
- ➔ Out: DOT graph

```
contract MyContract {
  uint balance;
  function MyContract() { Mint(1000000); }
  function Mint(uint amount) internal
    { balance = amount; }
  function Withdraw() { msg.sender.send(balance); }
  function GetBalance() constant returns(uint)
    { return balance; }
}
```

solgraph

github.com/raineorshine/solgraph



Anyone can run solgraph to **see potential security risks** in a smart contract.

Dynamic Analysis

Dynamic analysis is a method of testing and evaluating a program by executing its code.

We need standardized unit testing patterns

- Access Control
- NoEther
- ...

And now something non-technical.

But important.

The Three Developer Cultures

The Three Developer Cultures

Web Developer

JS, Java, PHP, Ruby, Python

Values simplicity, usability, practicality.

Doesn't intuit systems level pitfalls.

"It works for me!"

The Three Developer Cultures

Web Developer

JS, Java, PHP, Ruby, Python

Values simplicity, usability, practicality.

Doesn't intuit systems level pitfalls.

“It works for me!”

Systems Engineer

C++, EVM Assembly

Understands system pitfalls

Undervalues abstraction

“I know every system quirk that could cause be a security concern!”

The Three Developer Cultures

Web Developer

JS, Java, PHP, Ruby, Python

Values simplicity, usability, practicality.

Doesn't intuit systems level pitfalls.

“It works for me!”

Systems Engineer

C++, EVM Assembly

Understands system pitfalls

Undervalues abstraction

“I know every system quirk that could cause be a security concern!”

Academic

F*, Why3

Rigorous solutions

Impractical (sometimes)

“We must be able to prove that it is secure!”

The ~~Three~~ 4 Developer Cultures

Non-Developer

Word, Mailchimp, Slack

No ability to distinguish the difficult from the trivial.

Source of speculation.

“How bad is it?”

Let’s work together.

Summary

- → **Use static analysis to detect code smells**
  e.g. solgraph
- → **Use dynamic analysis**
  Unit testing patterns needed
- → **The 3 (+1) Developer Cultures**
  Evolve in the right direction

Thank you

Raine Rupert Revere
github raineorshine
twitter @metaraine