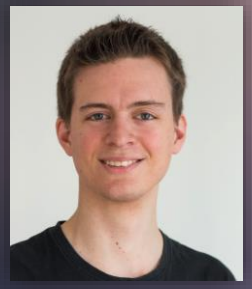
SRL
SOFTWARE RELIABILITY LAB

ETH zürich

SECURIFY

Not Your Grandma's Smart Contract Verification


Florian Buenzli


Dana Drachsler-Cohen


Andrei Dan


Arthur Gervais


Quentin Hibon

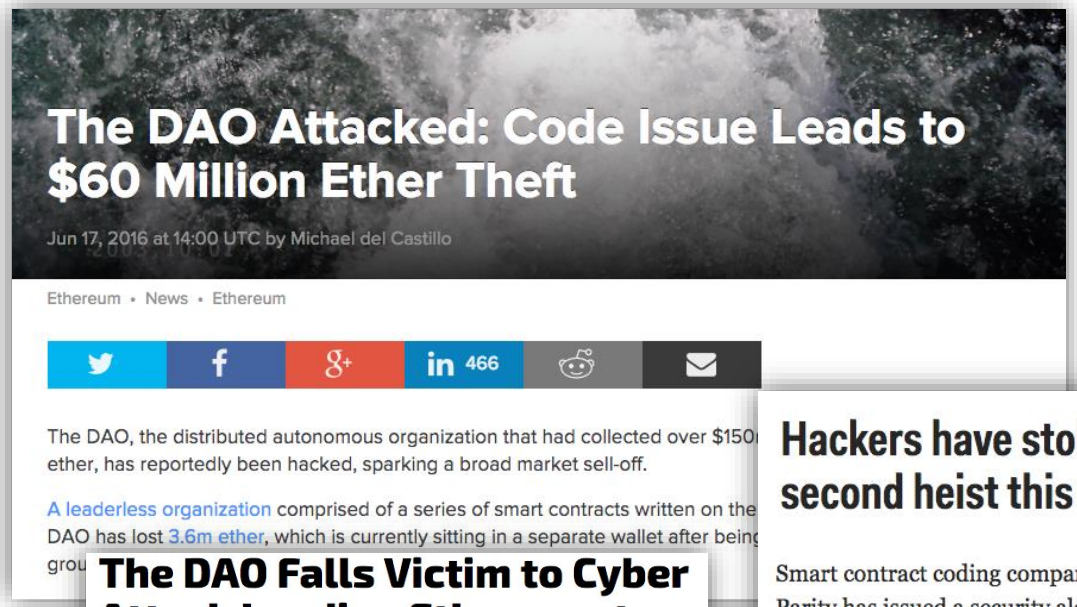

Hubert Ritzdorf


Petar Tsankov


Martin Vechev

http://blockchainsecurity.ethz.ch

Smart Contract *Security Bugs* in the News



Unprivileged write to storage

Any user may change the wallet's owner

Wallet Contract

```
address owner = ...;

function initWallet(address _owner) {
  owner = _owner;
}

function withdraw(uint amount) {
  if (msg.sender == owner) {
    owner.transfer(amount);
  }
}
```

Only owner can send ether

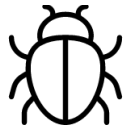
An attacker used a similar bug to ***steal $30M*** in July

More Security Bugs…

Unexpected ether flows

More Security Bugs…

Unexpected ether flows



Insecure coding, such as unprivileged writes *(e.g., Multisig Parity bug)*

More Security Bugs…

Unexpected ether flows

Insecure coding, such as unprivileged writes *(e.g., Multisig Parity bug)*

Use of unsafe inputs (e.g., reflection, hashing, …)

More Security Bugs…

Unexpected ether flows

Insecure coding, such as unprivileged writes *(e.g., Multisig Parity bug)*

Use of unsafe inputs (e.g., reflection, hashing, …)

Reentrant method calls *(e.g., DAO bug)*

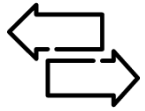
More Security Bugs…

Unexpected ether flows

Insecure coding, such as unprivileged writes *(e.g., Multisig Parity bug)*

Use of unsafe inputs (e.g., reflection, hashing, …)

Reentrant method calls *(e.g., DAO bug)*



Manipulating ether flows via transaction reordering

Transaction *reordering*

The owner can change the price

Token Contract

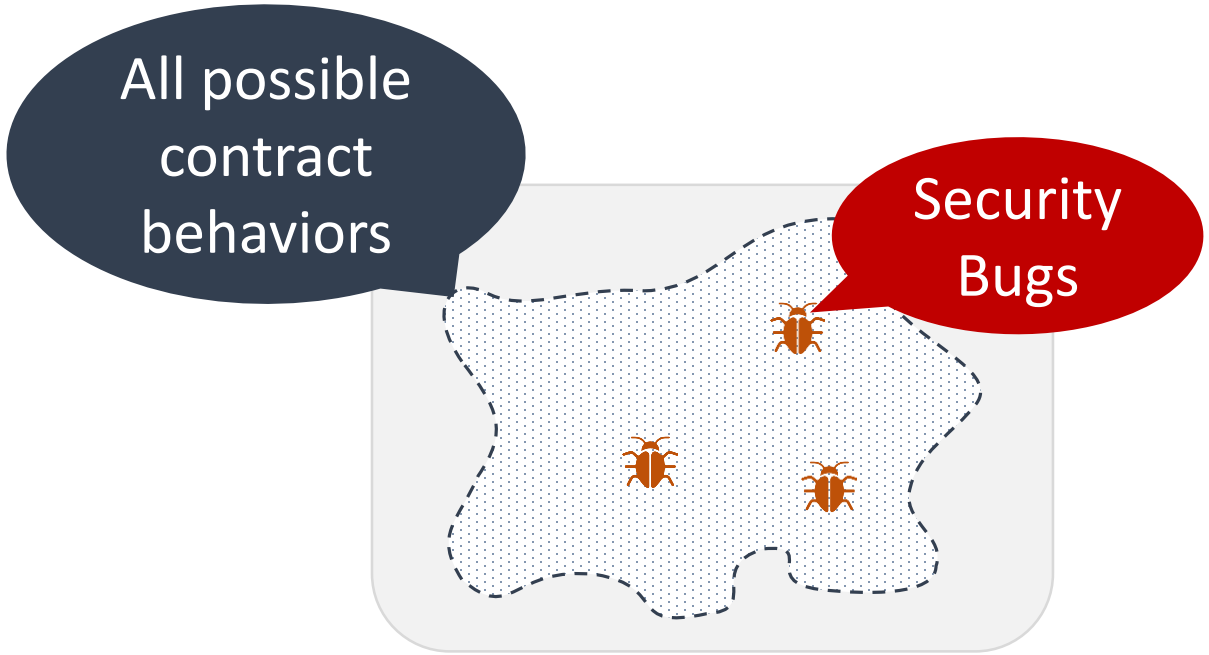
```
uint price = 10;
address owner;

function setPrice(uint newPrice) {
    if (msg.sender == owner)
      price = newPrice;
  }

function sellToken() {
    msg.sender.transfer(price);
}
```

A user can buy with the current price

The two operations do not commute

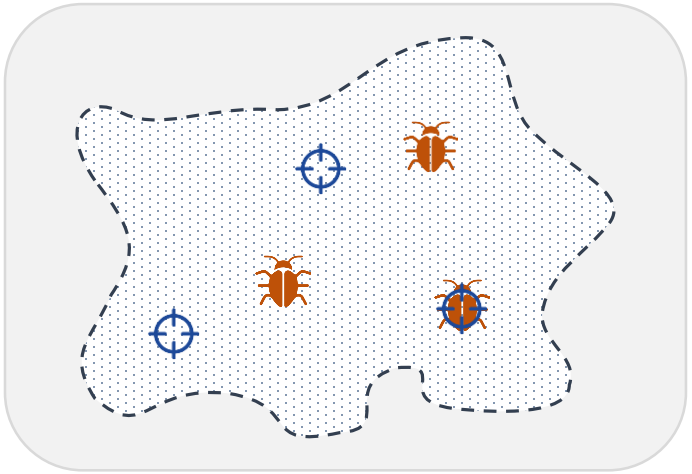

Automated Security Analysis

Automated Security Analysis Approaches



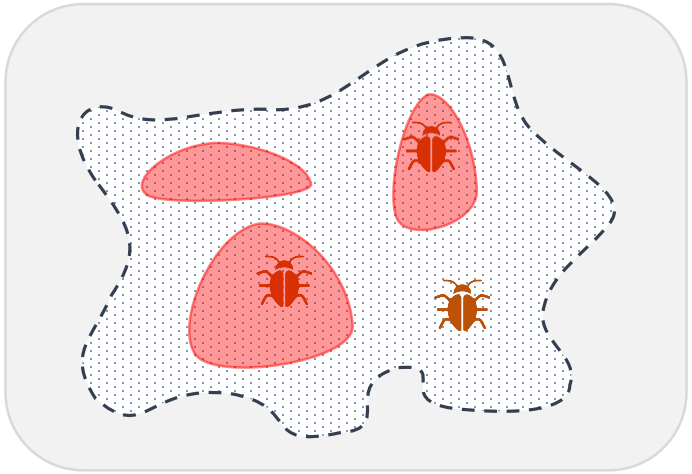
Problem: Cannot enumerate all possible contract behaviors…

Security Analysis Approaches



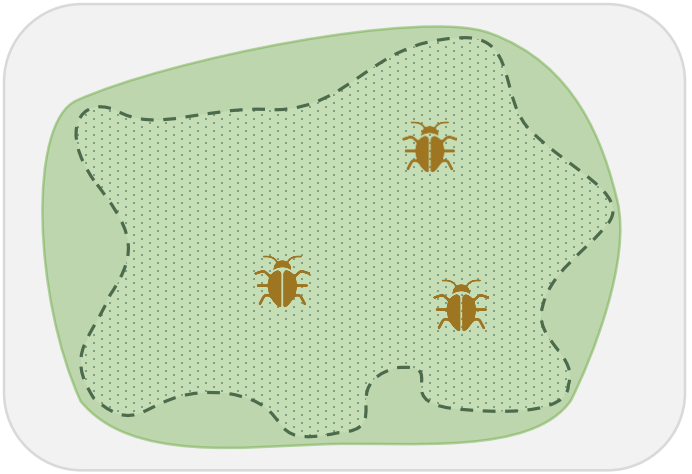
Testing

Report true bugs
Can miss bugs



Dynamic (symbolic) analysis

Report true bugs
Can miss bugs



Automated verification

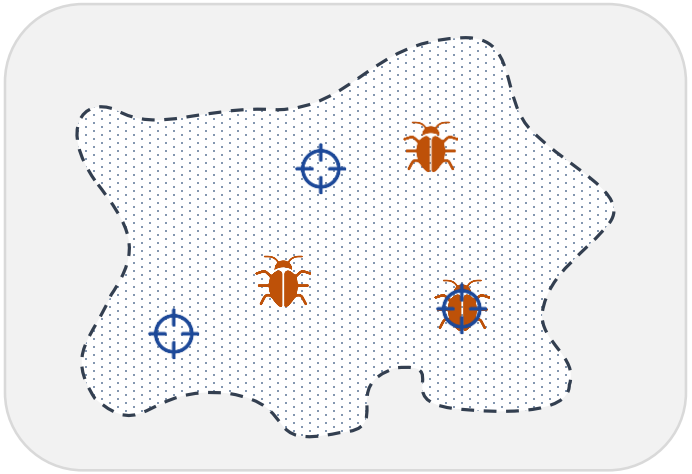
Can report false alarms
No missed bugs

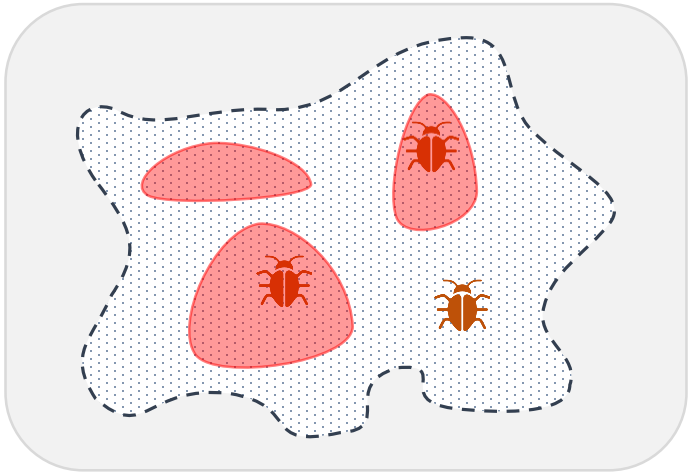
Current State of Automated Analysis for Ethereum Smart Contracts

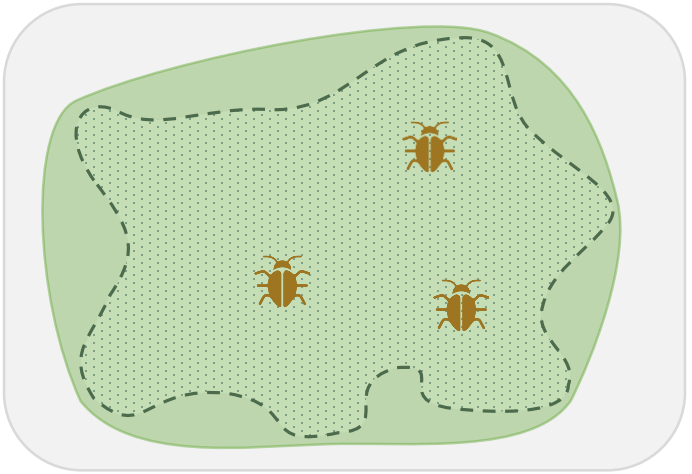
Security Analysis Approaches

Populus

Oyente







Testing

Dynamic (symbolic) analysis

Automated verification

Report true bugs
Can miss bugs

Report true bugs
Can miss bugs

Can report false alarms
No missed bugs

SECURIFY

www.securify.ch

Fully ***automated***, one-click, ***formal verification*** system for Ethereum smart contracts


Demo

Securify: Under the Hood

Captures key semantic facts about the contract

Security patterns expressed in a ***designated security language***

```
00: 60
02: 5b
04: 42
06: 80
08: 90
0a: 56
 ⋮
```

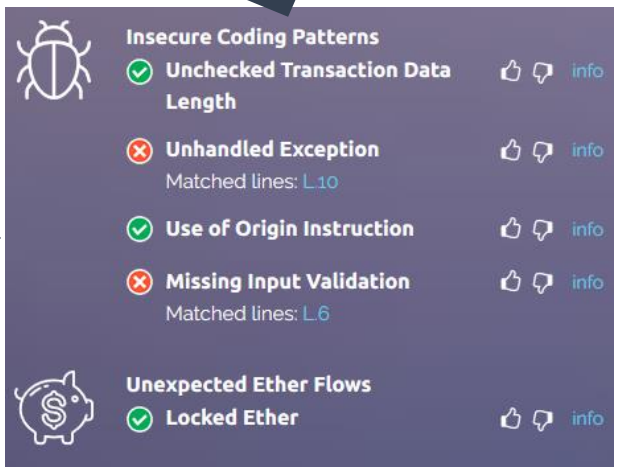
Decomp.

```
00: x = Balance
02: y = 0x20
04: If (x == 0x00)
06:  MStore(y, x)
08:  z = y
0a:  goto 0x42
        ⋮
```

Static Analysis

```
MemTag(0x20, Balance)
MemTag(0x40, Const)
VarTag(z, Const)
VarTag(k, Gas)
Assign(s, 0x20)
Call(s{0x20}, k{Gas})
        ⋮
```

Infer



EVM Binary

Securify Intermediate Representation

Securify Semantic Representation

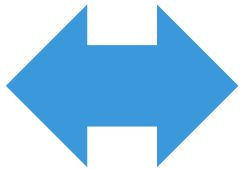
Securify Report

Fully automated, easily extensible

ChainSecurity

Swiss-based startup that provides intelligent security solutions for ***blockchains*** and ***smart contracts***

Automated
Security Analysis Systems



Comprehensive Smart Contract Auditing

https://chainsecurity.com

Summary

Research

Fully automated

Strong guarantees

Extensible

https://www.securify.ch

Product

Get in touch with our team of ***security*** / ***blockchain*** / ***program analysis*** experts

https://chainsecurity.com

contact@chainsecurity.com

@chain_security