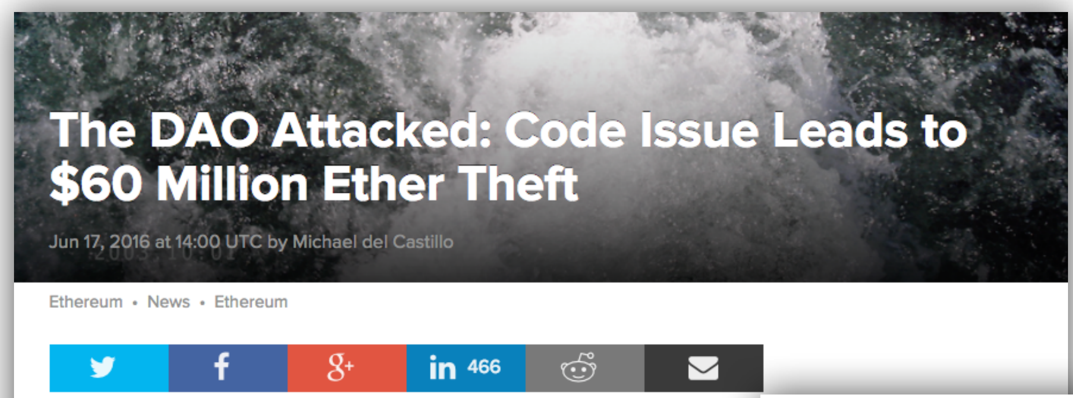
Smart Contract *Security Bugs* in the News



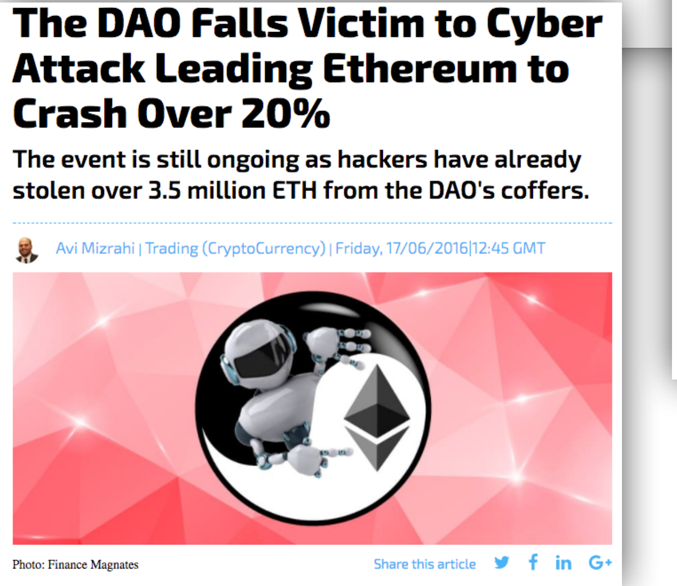
The DAO Attacked: Code Issue Leads to $60 Million Ether Theft

Jun 17, 2016 at 14:00 UTC by Michael del Castillo

Ethereum • News • Ethereum

The DAO, the distributed autonomous organization that had collected over $150
ether, has reportedly been hacked, sparking a broad market sell-off.

A leaderless organization comprised of a series of smart contracts written on the
DAO has lost 3.6m ether, which is currently sitting in a separate wallet after being
grou

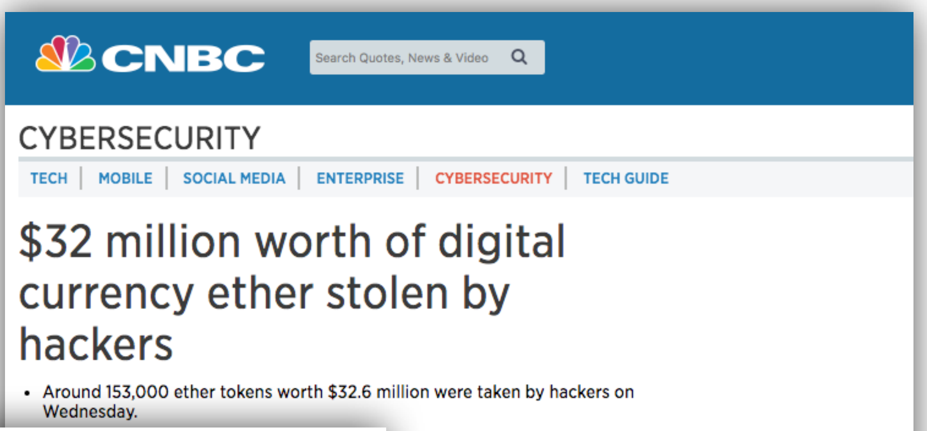


The DAO Falls Victim to Cyber Attack Leading Ethereum to Crash Over 20%

The event is still ongoing as hackers have already stolen over 3.5 million ETH from the DAO's coffers.

Avi Mizrahi | Trading (CryptoCurrency) | Friday, 17/06/2016|12:45 GMT

Photo: Finance Magnates



CNBC
Search Quotes, News & Video

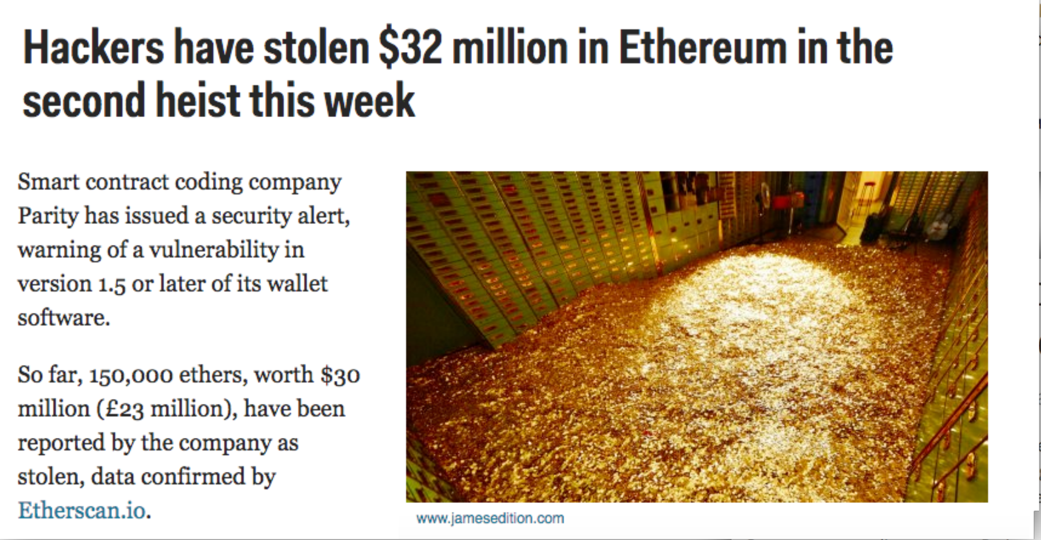
CYBERSECURITY

TECH | MOBILE | SOCIAL MEDIA | ENTERPRISE | CYBERSECURITY | TECH GUIDE

$32 million worth of digital currency ether stolen by hackers

- Around 153,000 ether tokens worth $32.6 million were taken by hackers on Wednesday.
- llet was exploited by hackers.
- day where $7 million worth of ether

ET Thu, 20 July 2017

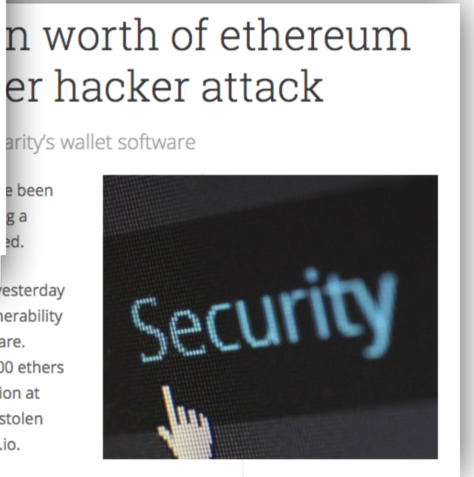


Hackers have stolen $32 million in Ethereum in the second heist this week

Smart contract coding company Parity has issued a security alert, warning of a vulnerability in version 1.5 or later of its wallet software.

So far, 150,000 ethers, worth $30 million (£23 million), have been reported by the company as stolen, data confirmed by Etherscan.io.

www.jamesedition.com



n worth of ethereum
er hacker attack

arity's wallet software

e been
g a
ed.

Smart contract coding company Parity yesterday issued a security alert, warning of a vulnerability in version 1.5 or later of its wallet software. According to the company, so far 150,000 ethers have been stolen, worth nearly $35 million at current price levels. The amount of the stolen ether has been confirmed by Etherscan.io.

What are Ethereum Smart Contracts?

```
contract Wallet {
  uint balance = 10;

  function withdraw(){
    if(balance > 0)
      msg.sender.call.value(balance)();
    balance = 0;
} }
```

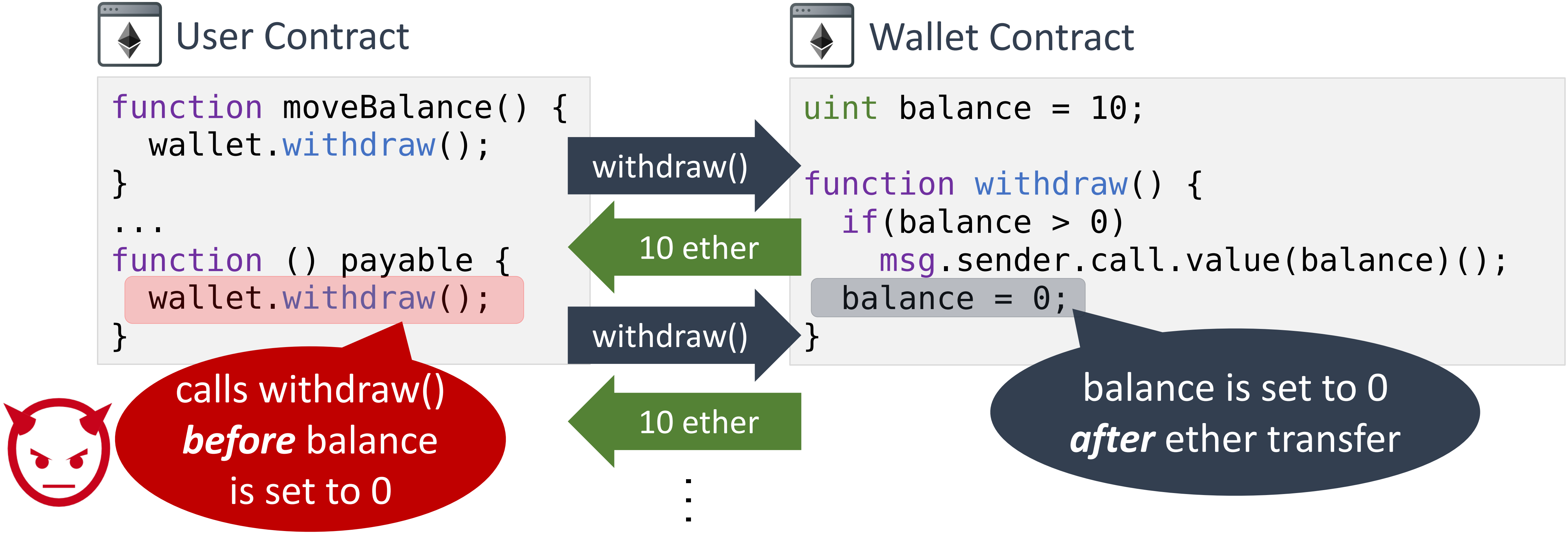
Transfer $$$ to the caller

- Small programs that ***handle money*** (ether)
- Executed on the Ethereum blockchain
- Written in high-level languages (*e.g.*, Solidity)
- ***No patching*** after release

What can go wrong when programs handle billions of USD?

Security Bugs in Ethereum Smart Contracts

Security Bug #1: Reentrancy



User Contract

```
function moveBalance() {
  wallet.withdraw();
}
...
function () payable {
  wallet.withdraw();
}
```

Wallet Contract

```
uint balance = 10;

function withdraw() {
  if(balance > 0)
    msg.sender.call.value(balance)();
  balance = 0;
}
```

calls withdraw() ***before*** balance is set to 0

balance is set to 0 ***after*** ether transfer

An attacker used this bug to steal 3.6M ether (equivalent of ***$1B today***)

Security Bug #2: *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.send(amount);
  }
}
```

Only owner can send ether

An attacker used a similar bug to ***steal $32M*** few weeks ago

More Security Bugs…

Unexpected ether flows

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

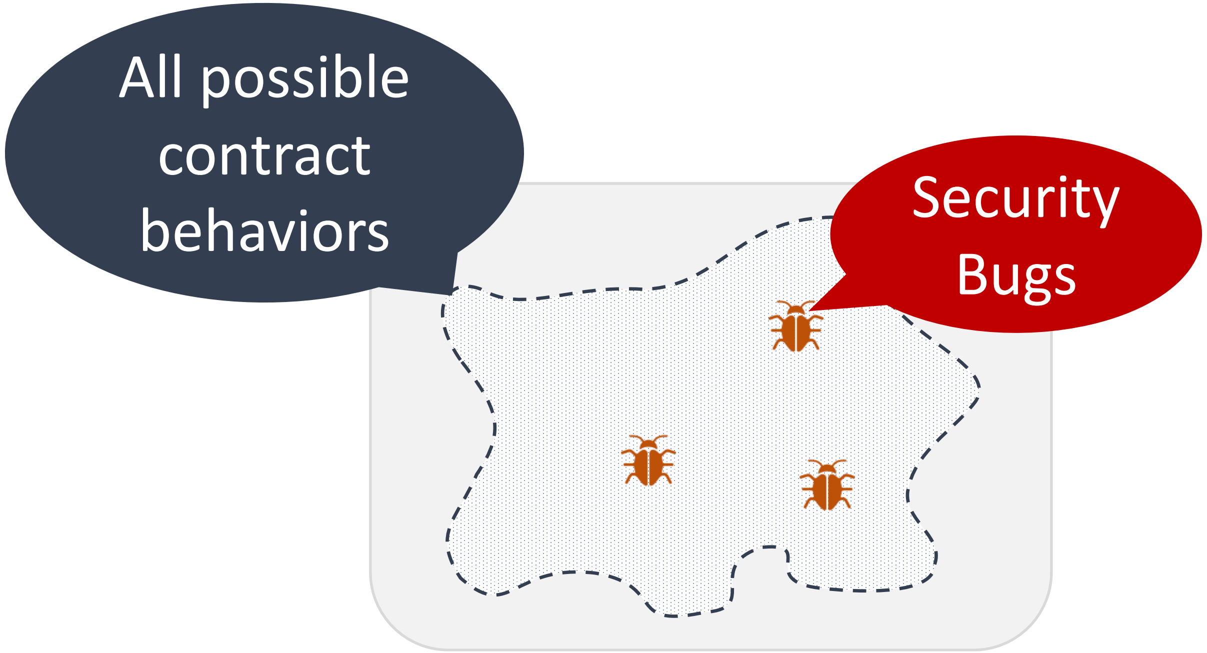
Use of unsafe inputs (e.g., reflection, hashing, …)

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

Transaction reordering

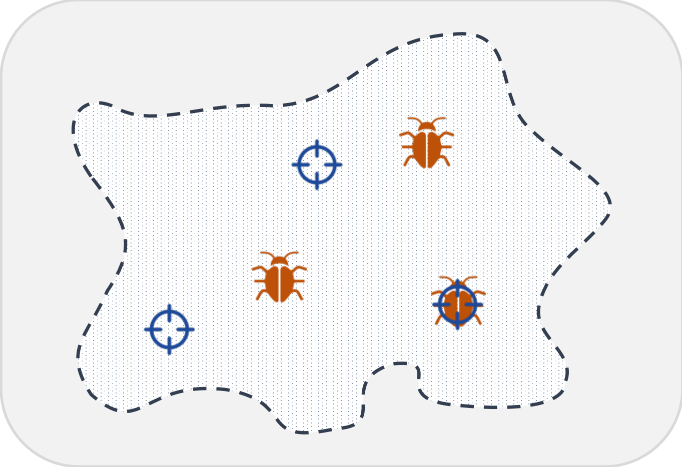
Automated Security Analysis

Automated Security Analysis: Existing Solutions



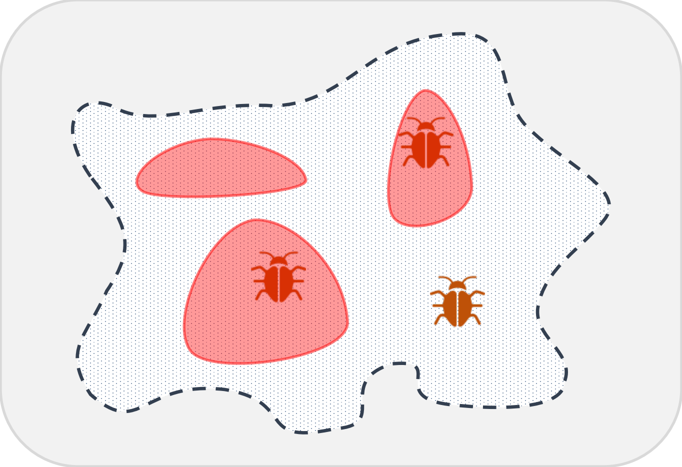
Problem: Cannot enumerate all possible contract behaviors…

Automated Security Analysis: Existing Solutions



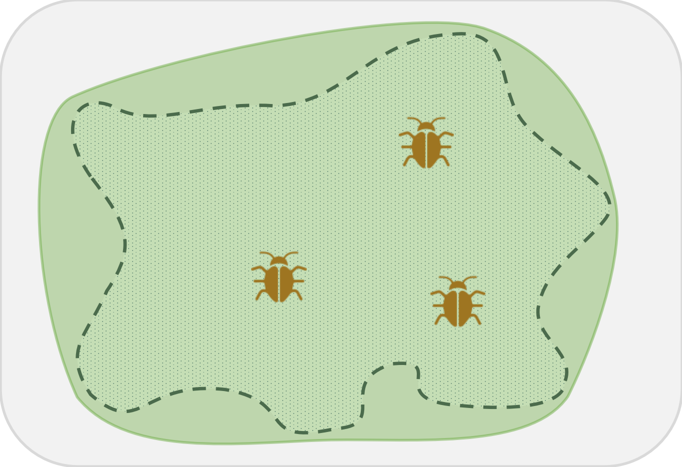
- Testing

Very limited guarantees



- Dynamic analysis
- Symbolic execution

Better than testing, but
can still miss vulnerabilities



- Static analysis
- Formal verification

Strong guarantees

SECURIFY

www.securify.ch

The first fully ***automated***, one-click, ***formal verification system*** for Ethereum smart contracts

Provides ***trust*** towards both contract users and developers