Безопасность ICO контрактов (7)

Александр Половьян
alex@ledgers.world

batchOverflow

- https://medium.com/@peckshield/alert-new-batchoverflow-bug-in-multiple-erc20-smart-contracts-cve-2018-10299-511067db6536

- Классика переполнения

Haha, classic

```
function batchTransfer(address[] _rs, uint256 _v) … {
    …
    uint256 amount = uint(_rs.length) * _v;
    balances[msg.sender] = balances[msg.sender] - amount;
    …
}
```

Лекарство

- Ad-hoc проверка:
  *uint256 amount = uint(_rs.length) * _v;*
  require(amount > _v);

- Библиотеки с контролем переполнения

  https://github.com/OpenZeppelin/openzeppelin-solidity/blob/master/contracts/math/SafeMath.sol

  using SafeMath for uint;
  …
  uint256 amount = uint(_rs.length).mul(_v);

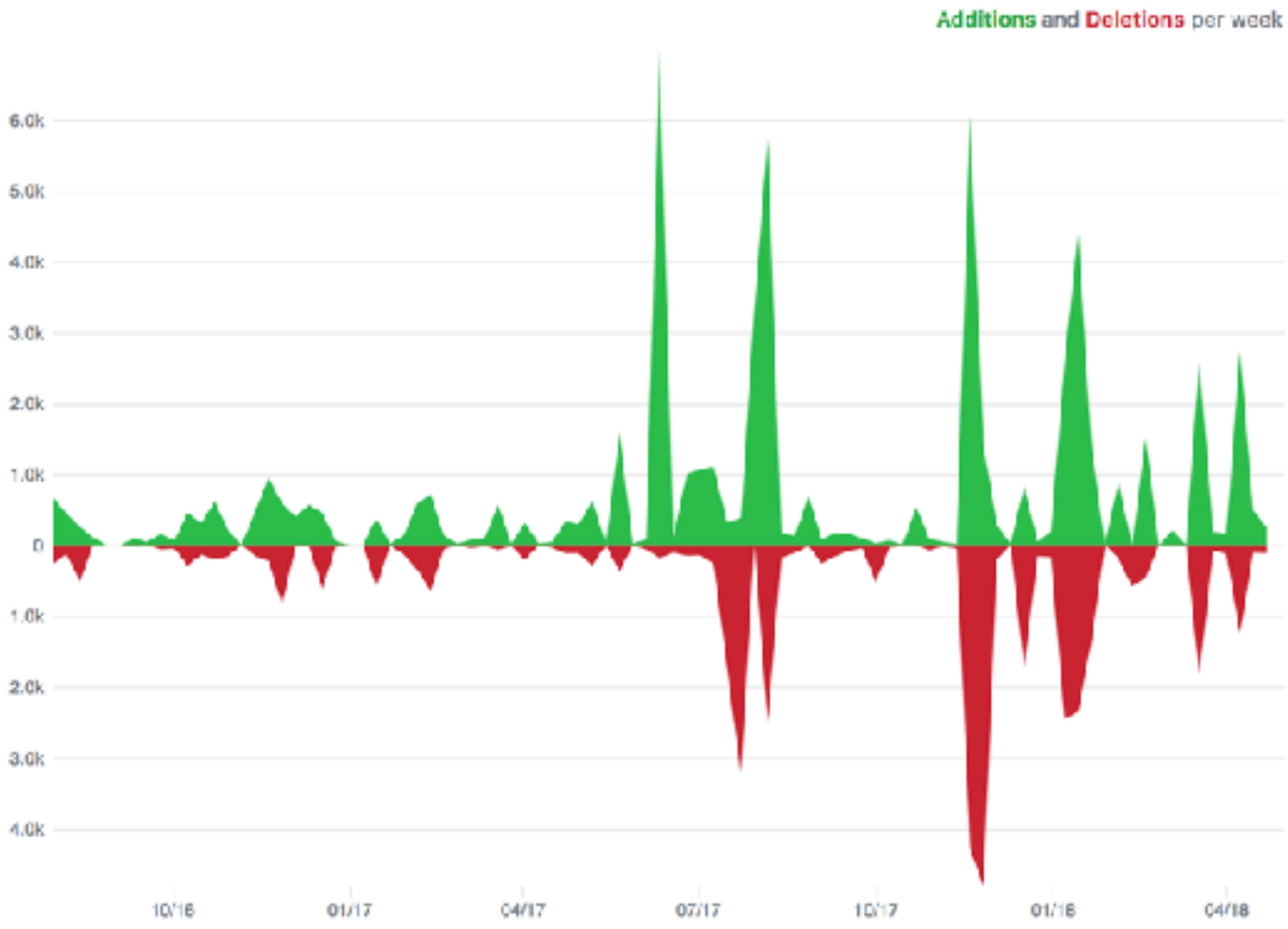
Стандартные реализации

- Вместе с EIP обычно публикуется код
- EIP не касается реализации, это только договоренность об интерфейсе
- Всегда проверяйте используемый код
- Библиотека решений zeppelin-solidity

zeppelin-solidity

- MIT license
- https://openzeppelin.org
- https://github.com/OpenZeppelin/zeppelin-solidity/
- ~$ git submodule add http://github.com/OpenZeppelin/zeppelin-solidity contracts/zeppelin-solidity

Статистика репозитория



What’s in the box?

Ownable

```
contract Ownable {
    address public owner;
    event OwnershipTransferred (…

    function Ownable() public { …

    modifier onlyOwner() { …

    function transferOwnership(address newOwner)
        public onlyOwner { …
}
```

Claimable

```
contract Claimable is Ownable {
    address public pendingOwner;

    modifier onlyPendingOwner() { …

    function transferOwnership(address newOwner)
        onlyOwner public { …

    function claimOwnership() onlyPendingOwner public {…
}
```

Contactable (зачем?)

```
contract Contactable is Ownable {
  string public contactInformation;

  function setContactInformation(string info) onlyOwner public {
    contactInformation = info;
  }

}
```

BasicToken

- Реализация ERC20Basic, ERC20Basic < EIP20

- function **totalSupply**() public view returns (uint256);

- function **balanceOf**(address who) public view returns (uint256);

- function **transfer**(address to, uint256 value)
  public returns (bool);

- event **Transfer**(address indexed from,
  address indexed to, uint256 value);

StandardToken

- StandardToken is ERC20, BasicToken

- function **allowance**(address owner, address spender)
  public view returns (uint256);

- function **transferFrom**(address from, address to, uint256 value)
  public returns (bool);

- function **approve**(address spender, uint256 value)
  public returns (bool);

- event **Approval**(address indexed owner,
  address indexed spender, uint256 value);

MintableToken

- function **mint**(address _to, uint256 _amount)
  onlyOwner canMint public
  returns (bool) {…

- function **finishMinting**()
  onlyOwner canMint public
  returns (bool) {…

CappedToken

- contract **CappedToken** is **MintableToken**
- function **mint**(address _to, uint256 _amount)
  onlyOwner canMint public returns (bool)
  {
  require(totalSupply_.add(_amount) <= cap);
  return super.mint(_to, _amount);
  }

Crowdsale

- function **Crowdsale**(
  uint256 _rate, address _wallet, ERC20 _token)
  public { …

- function **()** external payable { …

- function **buyTokens**(address _beneficiary)
  public payable { …

- event **TokenPurchase**(
  address indexed purchaser,
  address indexed beneficiary,
  uint256 value, uint256 amount);

На что обратить внимание?

- Не ownable
- Токен не является частью контракта
- Расширяется через переопределение внутренних функций

Как расширять?

- Финализация
- Временные ограничения
- Hard cap
  максимальное количество привлеченных средств
- Soft cap
  минимальное количество привлеченных средств
- Whitelist
- Повышающаяся цена