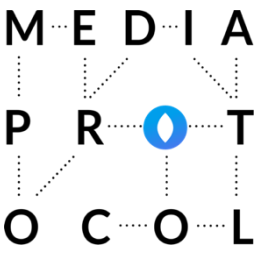
SmartDec

MEDIA Protocol Smart Contracts Security Analysis

This report is public.

Published: March 21, 2018



Abstract

In this report we consider the security of the MEDIA Protocol project. Our task is to find and describe security issues in the smart contracts of the platform.

Procedure

In our audit, we consider the following crucial features of the smart contract code:

1. Whether the code is secure.
2. Whether the code corresponds to the documentation (including whitepaper).
3. Whether the code meets best practices in efficient use of gas, code readability, etc.

We perform our audit according to the following procedure:

- automated analysis
  - we scan project's smart contracts with our own Solidity static code analyzer SmartCheck
  - we scan project's smart contracts with several publicly available automated Solidity analysis tools such as Remix, Oyente, and Solhint
  - we manually verify (reject or confirm) all the issues found by tools
- manual audit
  - we manually analyze smart contracts for security vulnerabilities
  - we check smart contracts logic and compare it with the one described in the whitepaper
  - we check ERC20 compliance
  - we run tests and check code coverage
- report
  - we reflect all the gathered information in the report

Disclaimer

The audit does not give any warranties on the security of the code. One audit can not be considered enough. We always recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. Besides, security audit is not an investment advice.

Checked vulnerabilities

We have scanned MEDIA Protocol smart contracts for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered (the full list includes them but is not limited to them):

- Reentrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with (Unexpected) Throw
- DOS with (Unexpected) revert
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed
- Redundant fallback function
- Send instead of transfer
- Style guide violation
- Unchecked external call
- Unchecked math
- Unsafe type inference
- Implicit visibility level
- Address hardcoded
- Using delete for arrays
- Integer overflow/underflow
- Locked money
- Private modifier
- Revert/require functions
- Using var
- Visibility
- Using blockhash
- Using SHA3
- Using suicide
- Using throw
- Using inline assembly

Project overview

In our analysis we consider MEDIA Protocol whitepaper (MEDIA Protocol Technical White Paper MASTER.docx (1).pdf, sha1sum 3d132737f415a1b0e8a49a9fc09b785560976047) and smart contracts code (media-token-master.zip, sha1sum 96b45bb9a0e9900754396abb285ba040867d15a4, contains git-repository, version on commit ce9e4aafb867b6ed6d395aa8758cc9fd2c9c22ef).

The latest version of the code

We have performed the check of the fixed vulnerabilities in the latest version of code — Git repository, version on commit efdcb7d, or MediaProtocol-master.zip, sha1sum d297f8da2f4efcac11133578fdba41e5a232d74e.

Project architecture

For the audit, we have been provided with the following set of files:

Source File Name	Contracts / Libraries	Description
ContentPurchaseTracker.sol	ContentPurchaseTracker contract ContentPurchaseTrackerDispatchercontract ContentPurchaseTrackerImplementation contract	Purchase content
Delegatable.sol	Delegatable contract DelegatableStorage contract DelegatableDispatcher contract DelegatableImplementation contract	Delegation support (Delegates can execute certain methods on behalf of the master contract)
IdentityVerification.sol	IdentityVerification contract IdentityVerificationStorage contract IdentityVerificationDispatcher contract IdentityVerificationImplementation contract	Identity verification services
InteractionsCounter.sol	InteractionsCounterStorage contract InteractionsCounter contract InteractionsCounterDispatcher contract InteractionsCounterImplementation contract	User interactions monitoring

MediaManager.sol	MediaManager contract	MediaToken menagement
MediaToken.sol	MediaToken contract	ERC20 token implementation
MediaTokenUser.sol	MediaTokenUser contract	Contains hardcoded token address
Migrations.sol	Migrations contract	Migrations
Promotion.sol	promotionLibrary library Promotion contract	Individual promotions handling
Promotions.sol	Promotions contract PromotionsStorage contract PromotionsDispatcher contract PromotionsImplementation contract	
StandardTokenWithRecurrency.sol	StandardTokenWithRecurrency contract	Extends ERC20 StandardToken, supports recurrent payments
SubscriptionManager.sol	SubscriptionDefinition contract SimpleSubscriptionDefinition contract subscriptionManagerLibrary library SubscriptionManager contract SubscriptionManagerStorage contract SubscriptionManagerDispatcher contract SubscriptionManagerImplementation contract	Subscriptions handling
Upgradeable.sol	Upgradeable contract UpgradeableDispatcher contract UpgradeableImplementer contract	Upgradeable contracts implementation

Code logic

The basis of the smart contract architecture is the upgradeable smart contract as implemented in Upgradeable.sol. The idea of upgradeable smart contract is that the contract is separated in two parts: UpgradeableDispatcher and UpgradeableImplementer.

- UpgradeableDispatcher
  - serves as a front-end proxy and forwards (delegates) all messages to the implementer of UpgradeableImplementer contract
  - all state variables are stored in the `dispatcher` contract
  - allows implementers to have access to the dispatcher's state variable through the `delegatecall` function
  - stores the address of the current implementer in `_implementer` state variable
  - current implementer may be replaced at any moment by calling the `replaceImplementation` function
  - after a new implementer is installed, its `initialize` method is invoked via `delegatecall` that means that the method gets all state variables of the `dispatcher` contract rather than the implementer contract
  - only the owner of the `dispatcher` smart contract may replace the implementation
  - is created with no owner (`_owner == 0`) set
  - the first call to `setOwner` function sets the owner
  - after the call the owner resetting is not possible.
- MediaToken
  - ERC20 compatible token with the following parameters:
    - Name: MediaToken
    - Symbol: MEDIA
    - Decimals: 18
  - supports recurrent payments
  - the owner may approve spending of some amount of tokens per specified `interval`
  - `interval` is measured in blocks
  - `approveRecurrent` function creates or updates the approval for a specified address
  - `canSpend` function checks if the given `spender` may spend the given `value` of tokens on behalf of the `from` address.
- MediaManager
  - is the managing contract
  - is created simultaneously with MediaToken
  - has the same owner as MediaToken
  - may perform token transfers on behalf of any user
  - a user may disable such transfers by calling `disableManager` function
  - once disabled there is no way to re-enable MediaManager services.

The central smart contract is PromotionsImplementation. It inherits all implementation contracts:

- InteractionsCounterImplementation
- SubscriptionManagerImplementation
- ContentPurchaseTrackerImplementation.

The proxy contract PromotionsDispatcher inherits:

- InteractionsCounterDispatcher
- SubscriptionManagerDispatcher
- ContentPurchaseTrackerDispatcher.

The constructor of PromotionsDispatcher sets the owner and the implementation for the dispatcher.
The following contracts are not intended for deployment but are inherited by other smart contracts:

- ContentPurchaseTracker
- ContentPurchaseTrackerDispatcher
- ContentPurchaseTrackerImplementation (that is whole ContentPurchaseTracker.sol)
- Delegatable
- DelegatableDispatcher
- DelegatableImplementation (Delegatable.sol)
- InteractionsCounterStorage
- InteractionsCounter
- InteractionsCounterDispatcher
- InteractionsCounterImplementation (InteractionsCounter.sol)
- MediaTokenUser (MediaTokenUser.sol)
- Promotion (Promotion.sol)
- Promotions, PromotionsStorage
- StandardTokenWithRecurrency
- SubscriptionManager
- SubscriptionManagerStorage
- SubscriptionManagerDispatcher
- SubscriptionManagerImplementation (SubscriptionManager.sol).

Deployment

The deployment is performed by script 2_deploy_contracts.js.
MediaToken smart contract is deployed with the initial balance of 1,000,000 MediaTokens.
MediaToken deploys MediaManager smart contract.
Then promotionLibrary and subscriptionManagerLibrary are deployed.
Then PromotionsImplementation is deployed followed by PromotionsDispatcher.
Finally SimpleSubscriptionDefinition is deployed.
IdentityVerification is not deployed by the script.

Automated analysis

We used several publicly available automated Solidity analysis tools.
Here are the combined results of SmartCheck, Solhint, and Remix. Oyente has found no issues.
All the issues found by tools were manually checked (rejected or confirmed).

Tool	Vulnerability	False positives	True positives
SmartCheck	Address Hardcoded	8	
	Constant Functions	2	
	Dos With Revert	30	
	Erc20 Approve	1	
	Erc20 Transfer Should Throw		1
	Functions Returns Type And No Return		1
	Gas Limit And Loops	2	8
	No Payable Fallback	12	26
	Pragmas Version	18	
	Private Modifier	10	
	Reentrancy External Call	53	1
	Revert Require	2	
	Tx Origin	3	
	Unchecked Call		1
	Unchecked Math	33	25
	Using Inline Assembly	1	1

	Var	1	
	Visibility	1	3
Total SmartCheck		177	67
Remix	Defines a return type but never explicitly returns a value		1
	Function state mutability can be restricted to pure	1	
	Gas requirement of function high	78	11
	Jump instructions and labels are low-level EVM features that can lead to incorrect stack access	2	
	No visibility specified		3
	Potential Violation of Checks-Effects-Interaction pattern	4	5
	Potentially should be constant but is not	7	1
	Return value of low-level calls not used		1
	The Contract uses inline assembly	2	
	Unused function parameter	2	
	Use of "delegatecall"	1	
	Use of selfdestruct	1	1
	Use of the "var" keyword is deprecated	3	
	Use of tx.origin	3	
	Variables have very similar names	7	
Total Remix		111	23

Solhint	Avoid to use inline assembly	2	
	Avoid to use low level calls	1	
	Code contains empty block	3	
	Compiler version must be fixed	18	
	Event and function names must be different	11	
	Explicitly mark visibility in function	1	2
	Explicitly mark visibility of state	1	
	Fallback function must be simple	1	
	fallback is not payable	1	
	Variable is unused	2	
Total Solhint		41	2
Total Overall		329	92

Cases when these issues lead to actual bugs or vulnerabilities are described in the next section.

Manual analysis

The contracts were completely manually analyzed, their logic was checked and compared with the one described in the documentation. Besides, the results of the automated analysis were manually verified. All confirmed issues are described below.

Critical issues

Critical issues seriously endanger smart contracts security. We highly recommend fixing them.

Unchecked call

In Upgradeable.sol, line 33, `delegatecall` is used without checking return value:
`_implementation.delegatecall(bytes4(keccak256("initialize()")));`
Expect calls to external contract to fail. When sending ether, check for the return value and handle errors. We recommend using `transfer` for ether transfers.
The issue has been fixed and is not present in the latest version of the code.

Medium severity issues

Medium issues can influence smart contracts operation in current implementation. We highly recommend addressing them.

Obsolete code

The dispatcher smart contract delegates actual function calls to the implementer using the default function handler feature of Solidity (Upgradeable.sol, line 59).
The default function handler obtains the signature of the function being called. The expected length of the return value of the function is expected to be stored in the `_returnSizes` map. This implies that prior to any call to the `dispatcher` is possible, the implementer must setup `_returnSizes` map for all functions that may be called via the `dispatcher`. This operation should be performed by `initialize()` function, which is called by the `dispatcher` when the implementation is installed by `replaceImplementation` function.
Note, however, that the current version of EVM supports RETURNDATASIZE and RETURNDATACOPY instructions. These instructions make return value size bookkeeping in `_returnSizes` map unnecessary.
We recommend upgrading the dispatcher contract source code.
The issue has been fixed and is not present in the latest version of the code.

Code logic error

There are several logic errors:

- `canSpend` function logic error
  If the current interval is over, i.e. the following condition in MediaToken.sol, line 70, is true:
  ```
  ( recurrentAllowed[from][spender].interval > 0 &&
  block.number >= recurrentAllowed[from][spender].intervalStart
  + recurrentAllowed[from][spender].interval )
  ```
  new `intervalStart` is computed as
  `newIntervalStart = (block.number / interval) * interval + intervalStart % interval`
  Let `block.number` be 1202, `interval` be 50, `intervalStart` be 1005. Then, according to the formula, we obtain `newIntervalStart` as 1205, that is in the future to the current block number (1202). It looks that the formula should use block number difference between the current block and the last `intervalStart`, i.e. as follows:
  `newIntervalStart = (block.number - intervalStart) / interval * interval + intervalStart`
  The issue has been fixed and is not present in the latest version of the code.
- MediaManager contract logic error
  Once a user opted-out from MediaManager services, there is no way to opt-in. Perhaps the contract owner should be able to reset the user's state.
  Comment from the developer:
  "This behavior is intentional. The use of MediaManager is temporary, and is needed to support hybrid server till the environment is ready to switch to fully-decentralised solution. Of course, the MediaManager by design is weakening the MediaToken and Mediatoken holders security, so anyone can opt-out some of their account (e.g. the accounts where large amount of tokens is held), while using the hybrid solution from other accounts. The opt-out is permanent. Allowing status reset by the admin, or accidentally by the user is exactly what we wanted to disable. Documentation updated."

Out of gas

The default function handler reserves 10000 gas for its own execution before calling the delegate. However, check that the available amount of gas exceeds 10000 is not performed, so unsigned integer underflow error is possible. The `delegate` function may be called with some huge amount of gas reservation.
We recommend adding `require(gas > 10000)` check.
The issue has been fixed and is not present in the latest version of the code.

Gas limit and loops

The Promotion contract functions contain loops with unknown number of iterations at the following lines:

- 174
  The issue has been fixed: the loop's length was limited to 10.
- 214
  The issue has been fixed: the loop was removed.
- 219
  Comment from the developer:
  "This loops frees some gas with the delete call, that covers the gas consumed in the given iteration."
- 237
  The issue has been fixed: the loop was removed.
- 243
  Comment from the developer:
  "This loops frees some gas with the delete call, that covers the gas consumed in the given iteration."
- 283
  The issue has been fixed: the visibility of method getReferral was changed to private and the loop's length was limited to 4.
- 295
  The issue has been fixed: the loop's length was limited to 4.
- 409
  *The issue has been fixed: the loop's length was limited to 10.*If the arrays are large enough, the functions will not fit in the block gas limit and the transactions calling it will thus never be confirmed. If the array can be influenced by an attacker (e.g., if an attacker can register arbitrary number of investor accounts), this can lead to an attack.

We highly recommend avoiding loops with big or unknown number of steps.

Unchecked math

Solidity is prone to an integer over- and underflow. Overflow leads to unexpected effects and can lead to loss of funds if exploited by malicious account. It is recommended to use the SafeMath library for all arithmetic operations. It is not used in

- Promotion.sol, line 253
  ```
  return ((what + rewardPeriod - 1) / rewardPeriod) *
  rewardPeriod;
  ```
  The issue has been fixed: `require()` was added.
- Promotion.sol, line 284
  ```
  value = value * (100 - p.referralDecrease) / 100;
  ```
  The issue has been fixed: `require(referralDecrease <= 100)` was added.
- Promotion.sol, line 349
  ```
  uint consumerBudget = (budget * (100 - referralShare)) / 100;
  ```
  The issue has been fixed: `require(referralShare <= 100)` was added.

Low severity issues

Low severity issues can influence smart contracts operation in future versions of code. We recommend to take them into account.

No return value

The `transfer` function signature contains the return value type in MediaToken.sol, line 125:

```
function transfer(address to, uint256 value) public returns (bool){
      if(super.transfer(to, value))
         ExternalTransfer(msg.sender, to, msg.sender, value);
   }
```

However, there is no return value in the function.
If you don't need the return value of the function, do not specify `returns` in the function signature
The issue has been fixed and is not present in the latest version of the code.

Transfer no throw

According to the ERC20 Token Standard, `transfer` should `throw` if the `_from` account balance does not have enough tokens to spend. However, in MediaToken.sol, line 125, `transfer` throws nothing.
We recommend following the ERC20 Token Standard.
The issue has been fixed and is not present in the latest version of the code.

Unchecked math

Solidity is prone to an integer over- and underflow. Overflow leads to unexpected effects and can lead to loss of funds if exploited by malicious account. We recommend using the SafeMath library for all arithmetic operations. It is not used in Promotion.sol, lines:

- 200
  The issue has been fixed and is not present in the latest version of the code.
- 204
  Comment from the developer:
  "Not an issue, as block number -1 cannot underflow"
- 211
  Comment from the developer
  "The if statement limits switchtime to be greater or equal to block number, and newLastProcessedBlock is blocknumber - {0,1} - this statement can hardly underflow."
- 218
  Comment from the developer:
  "The remainingBudgertAllocation is limited by total coins issued, and this is far from uint256 limit."
- 222
  The issue has been fixed: the checks were added.
- 223
  The issue has been fixed: the checks were added.
- 242
  The issue has been fixed: the checks were added.
- 253
  The issue has been fixed: the check was added.
- 261
  Comment from the developer:

"Allocation is limited by total coins issued, while referral share is less or equal to 100. The result this is far from uint256 limit."

- 264
  Comment from the developer:
  "Allocation is limited by total coins issued, while referral share is less or equal to 100. The result this is far from uint256 limit."
- 284
  The issue has been fixed and is not present in the latest version of the code.
- 349
  The issue has been fixed and is not present in the latest version of the code.
- 364
  The issue has been fixed and is not present in the latest version of the code.

Using inline assembly

Inline assembly is used in Upgradeable.sol, line 65:

```
assembly {
        // return _dest.delegatecall(msg.data)
            calldatacopy(0x0, 0x0, calldatasize)
            let ret := delegatecall(sub(gas, 10000), target, 0x0,
calldatasize, 0, len)
            jumpi(rev, iszero(ret))
            return(0, len)
            rev: revert(0,0)
        }
```

Inline assembly is a way to access the Ethereum Virtual Machine at a low level. This discards several important safety features of Solidity. We recommend not using `assembly` if possible.
The developer has updated code according to our recommendations, making this issue secure in current code implementation. However, in future implementations of the code this may lead to vulnerabilities, so we highly recommend paying attention to this pattern.

Implicit visibility level

Functions without a visibility modifier are public (i.e., can be called from outside the contract). Make sure this conforms to the intended functionality. Explicitly define function visibility levels (`public`, `private`, `external`, `internal`) to improve code readability (Migrations.sol, line 11, 15, 19)
The issue has been fixed and is not present in the latest version of the code.

Potential violation of Checks-Effects-Interaction pattern

There is a Checks-Effects-Interaction violation in:

- Promotion.sol, functions:
  - `getInteraction`
  - `processBacklog`
  - `processReferralBacklog`
- subscriptionManager.sol, functions:

- `subscribe`
- `renewSubscription`

In this cases the violations do not lead to actual vulnerabilities. However, we highly recommend following best practices including Checks-Effects-Interactions pattern since it helps to avoid many serious vulnerabilities.
Comment from the developer:
"All mentioned cases are trade-offs between user experience (near-real-time payouts, requiring to do transfers in a cycle and cost of transaction). Since the only contracts called are the one in our package (control), it is not an issue."

Use of `selfdestruct`

`selfdestruct` is used in Promotion.sol, line 274
```
selfdestruct(p.provider);
```
Using `selfdestruct` may unexpectedly block calls. We recommend being especially careful if this contract is planned to be used by other contracts (for example, contracts with the library, interactions). Selfdestruction of the contract can leave callers in an inoperable state.
Comment from the developer:
"The selfdestruct is called only once the validity of an individual promotion ends, and after proper clean-up. Any call after the validity and cleanup don't make sense anyway."

Hardcoded address

MediaTokenUser.sol contains hardcoded address:
```
address internal _token =
0x18D6B0208E0425eCe4a046f59342050af7CCBA97;
```
This address is not currently used on the Ethereum network. We highly recommend checking all the addresses before the deploy.
The issue has been fixed and is not present in the latest version of the code.

Codestyle issues

The MEDIA Protocol smart contracts contain the following codestyle issues:

- ContentPurchaseTracker contract does not define state variables or implements function. It should be declared as interface.
  The issue has been fixed and is not present in the latest version of the code.
- Delegatable contract meets all requirements of solidity interfaces. It should be declared as interface.
  The issue has been fixed and is not present in the latest version of the code.
- IdentityVerification contract should be declared as interface.
  The issue has been fixed and is not present in the latest version of the code.
- SubscriptionDefinition contract should be declared as interface.
  The issue has been fixed and is not present in the latest version of the code.
- SubscriptionManager contract should be declared as interface.
  The issue has been fixed and is not present in the latest version of the code.

Conclusion

In this report we have considered the security of MEDIA Protocol smart contracts. We performed our audit according to the procedure described above.
Our initial audit showed two critical vulnerabilities and several medium and low severity issues. We have discussed the results of the audit with MediaProtocol team. In cases when the developer was aware about possible issues and they were the part of the design we supplied the report with the developer's comments. All the other issues have been fixed by the developer.
The latest version of the code does not contain confirmed security issues.

This analysis was performed by SmartDec.

Alexandr Chernov, Chief Research Officer
Katerina Troshina, Chief Executive Officer
Elizaveta Kharlamova, Analyst
Ivan Ivanitskiy, Chief Analytics Officer
Igor Sobolev, Analyst
Alexander Seleznev, Chief Business Development Officer

Sergey Pavlin, Chief Operating Officer 

March 21, 2018

Appendix

Tests output

```
  Contract: MediaTokenTest
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x09b4f4570606336ac65ff022551377db42379b03 by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 1e+23
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x09b4f4570606336ac65ff022551377db42379b03 amount: 1e+23
Standard transfer, from: 0x09b4f4570606336ac65ff022551377db42379b03
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 5e+22
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x12f2d2c3a184292cd8a8698e3f8d7e0fd0d4a201 by:
0x12f2d2c3a184292cd8a8698e3f8d7e0fd0d4a201 amount: 100000000
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x12f2d2c3a184292cd8a8698e3f8d7e0fd0d4a201 amount: 100000000
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x7caad93748064216ce50afc7d571df58ae4aad98 by:
0x7caad93748064216ce50afc7d571df58ae4aad98 amount: 100
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x7caad93748064216ce50afc7d571df58ae4aad98 by:
0x7caad93748064216ce50afc7d571df58ae4aad98 amount: 100
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x7caad93748064216ce50afc7d571df58ae4aad98 amount: 100
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x7caad93748064216ce50afc7d571df58ae4aad98 amount: 100
✓ Shall create token contract and transfer 100000000000000000000000
tokens to other user (3254ms)

Contract: PromotionsTest
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0xc929538032407e2606acce6ea6e0345bfe324686 by:
0x7caad93748064216ce50afc7d571df58ae4aad98 amount: 100
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0xc929538032407e2606acce6ea6e0345bfe324686 amount: 100
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x3a22cc352efa888879a8868a701beae2e0aedda2 by:
0x87863b8e40cb94c7b37a5f9160a72a2be35510a4 amount: 2000000
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 2000000
```

```
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x3a22cc352efa888879a8868a701beae2e0aedda2 by:
0x87863b8e40cb94c7b37a5f9160a72a2be35510a4 amount: 200000
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 200000
External transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 by:
0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 50000
External transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x09b4f4570606336ac65ff022551377db42379b03 by:
0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 50000
External transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x09b4f4570606336ac65ff022551377db42379b03 by:
0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 100000
Standard transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 50000
Standard transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x09b4f4570606336ac65ff022551377db42379b03 amount: 50000
Standard transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x09b4f4570606336ac65ff022551377db42379b03 amount: 100000
External transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 100000
Standard transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 100000
External transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x12f2d2c3a184292cd8a8698e3f8d7e0fd0d4a201 by:
0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 58332
External transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x7caad93748064216ce50afc7d571df58ae4aad98 by:
0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 58332
External transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0xc929538032407e2606acce6ea6e0345bfe324686 by:
0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 116666
Standard transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x12f2d2c3a184292cd8a8698e3f8d7e0fd0d4a201 amount: 58332
Standard transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x7caad93748064216ce50afc7d571df58ae4aad98 amount: 58332
Standard transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0xc929538032407e2606acce6ea6e0345bfe324686 amount: 116666
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
```

```
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
External transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0xad15d024266471cebb24449bf14df6a335965587 by:
0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 233334
External transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x8b2585b18a0962443eed5cdfd58e01aa3bacb25f by:
0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 233334
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
Standard transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0xad15d024266471cebb24449bf14df6a335965587 amount: 233334
Standard transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x8b2585b18a0962443eed5cdfd58e01aa3bacb25f amount: 233334
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
External transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x80df5362ae3a426e9d1ad4fae08928e8ca4266f9 by:
0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 20000
```

```
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
Standard transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x80df5362ae3a426e9d1ad4fae08928e8ca4266f9 amount: 20000
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
BigNumber { s: 1, e: 23, c: [ 9999999999, 99999997849840 ] }
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
External transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e by:
0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 10
External transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0xac48cfeac5853b94843fb7ca98654d4fa7062a95 by:
0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 80002
External transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x09b4f4570606336ac65ff022551377db42379b03 by:
0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 1100000
External transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 by:
0x3a22cc352efa888879a8868a701beae2e0aedda2 amount: 0
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
```

```
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
Standard transfer, from: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333
to: 0x69e5db7ce4488a4279027886f41cd2795a3dca8e amount: 10
Standard transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0xac48cfeac5853b94843fb7ca98654d4fa7062a95 amount: 80002
Standard transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x09b4f4570606336ac65ff022551377db42379b03 amount: 1100000
Standard transfer, from: 0x3a22cc352efa888879a8868a701beae2e0aedda2
to: 0x53aaaf9f3bc1dfe429ba2e72a9d47be018f1f333 amount: 0
BigNumber { s: 1, e: 6, c: [ 1250000 ] }
BigNumber { s: 1, e: 5, c: [ 100160 ] }
BigNumber { s: 1, e: 4, c: [ 58332 ] }
BigNumber { s: 1, e: 4, c: [ 58332 ] }
BigNumber { s: 1, e: 5, c: [ 116666 ] }
BigNumber { s: 1, e: 5, c: [ 233334 ] }
BigNumber { s: 1, e: 5, c: [ 233334 ] }
BigNumber { s: 1, e: 4, c: [ 20000 ] }
BigNumber { s: 1, e: 4, c: [ 80002 ] }
✓ Shall create new promotion and create some events in it (6765ms)
{ tx:
'0xf09698ca45c2d31847715e6edc06a26defef11f1437ef4f7becfb1dc60cb33dc'
,
receipt:
{ transactionHash:
'0xf09698ca45c2d31847715e6edc06a26defef11f1437ef4f7becfb1dc60cb33dc'
,
transactionIndex: 0,
blockHash:
'0x25e0514e3a538052d5f95cd0c7e9504077a6e5e51a1e0e96863fe86d0028e134'
,
blockNumber: 62,
gasUsed: 1420451,
cumulativeGasUsed: 1420451,
contractAddress: null,
logs: [ [Object], [Object], [Object] ],
status: 1 },
logs:
[ { logIndex: 2,
transactionIndex: 0,
transactionHash:
'0xf09698ca45c2d31847715e6edc06a26defef11f1437ef4f7becfb1dc60cb33dc'
,
```

```
blockHash:
'0x25e0514e3a538052d5f95cd0c7e9504077a6e5e51a1e0e96863fe86d0028e134'
,
blockNumber: 62,
address: '0x44c80fa620b130e8de16040967d78c4c433de52f',
type: 'mined',
event: 'CreatePromotion',
args: [Object] } ] }
✓ Shall allow actions only for verified users (2014ms)
✓ Shall allow actions on other account behalf (1000ms)
Subscribe, url: http://mypaywall by:
0x69e5db7ce4488a4279027886f41cd2795a3dca8e from: 88 to: 93
Subscribe, url: http://mypaywall by:
0x69e5db7ce4488a4279027886f41cd2795a3dca8e from: 95 to: 100
✓ Content buying and subscriptions shall work (2956ms)

5 passing (16s)
```