OBJECT COMPUTING

Object Computing, Inc.
12140 Woodcrest Executive Dr.
Suite 300
St. Louis, MO 63141
+1 314.579.0066

Alien Worlds daccustodian Contract Audit (EOS)

Source for this analysis is github: Alien-Worlds/contracts/daccustodian
Commit tag: 722e38267153bb53b37431aa86bb0959bd5485e7
Forked github: eosdac/eosdac-contracts
Forked Commit tag: f213b0f07c425912692c38e8b7e07ad99da9ddca

Files: contracts/daccustodian/daccustodian.[ch]pp
contracts/daccustodian/config.cpp
contracts/daccustodian/external_observable_actions.cpp
contracts/daccustodian/newperiod_components.cpp
contracts/daccustodian/pay_handling.cpp
contracts/daccustodian/paycpu.cpp
contracts/daccustodian/privatehelpers.cpp
contracts/daccustodian/registering.cpp
contracts/daccustodian/update_member_details.cpp
contracts/daccustodian/voting.cpp

Tests: contracts/daccustodian/daccustodian.test.ts
Ricardian contracts: contracts/daccustodian/daccustodian.contracts.md
Business requirements and other engineering artifacts:
https://eosdac.io/tools/smart-contracts-explained/
contracts/daccustodian/README.md

Executive Summary

No critical security violations were detected in this audit.

Remark	Minor	Major	Critical
5	0	0	0

Participants

Primary auditor is Jack DiSalvatore and Phil Mesnier reviewed.

Tools

Static analyzers

- EOSafe appears to be an Academic exercise last updated nearly 4 years ago
- EOSlime is a javascript tool that is also at least 2 years old
- Klevoya is a wasm analyzer tool using simple pattern matches

No static analyzers were used in this exercise. I tried to build the EOSafe tool, but it would not compile. The EOSlime tool was too complicated for me to determine how to build it. Klevoya was not useful because I could not generate an ABI file for the mining contract.

Known Vulnerabilities

List origin github:slowmisg/eos-smart-contract-security-best-practices/Readme_EN.md commit tag 5f77e19e50373d341e17a003c492388e9891a2c0

- **Numerical Overflow:** when doing token math, use the `asset` object operands and do not perform operations on the `uint256` that comes from `asset.amount`
- **Authorization Check:** make sure function parameters are consistent with the caller and use `require_auth` to check
- **Apply Check:** if the contract implements an `apply` function, bind each key action and code to meet the requirements, in order to avoid abnormal and illegal calls.
- **Transfer Error Prompt:** When processing a notification triggered by `require_recipient`, ensure that `transfer.to` is `_self`
- **Random Number Practice:** Random number generator algorithm should not introduce controllable or predictable seeds

We relied on visual inspection to look for instances of code that were similar to the examples found in the above document as well as best practices accumulated through many years of experience.

Findings And Recommendations

We examined 6 tables, 25 actions and 23 helper functions.

Remarks

- Mixing snake-case and camel case for function names.
- Use of magic numbers
- Possible divide by zero in newperiod_components.cpp#267
- Function implementation not found for `add_all_auths_msig`
- Function implementation not found for `validateUnstake`

Minor

-

Major

-

Critical

-

Table/Singleton	Notes
config2	Remark: use constants instead of magic numbers.
state	No issues.
votes	No issues.
proxies	No issues.
pendingpay	No issues.
candperms	No issues.

Action	Notes	Ricardian Contract
ACTION updateconfige(const contr_config &newconfig, const name &dac_id);	Remark: use constants instead of magic numbers. **Question**: line#38-39 does the current state only need to be set here if it doesn't already exist? Is this code needed?	Yes
ACTION balanceobsv(const vector<account_balance_delta> &account_balance_deltas, const name &dac_id);	No issues.	No
ACTION stakeobsv(const vector<account_stake_delta> &account_stake_deltas, const name &dac_id	No issues.	No

); 		

ACTION
weightobsv(
  const vector<account_weight_delta> &account_weight_deltas,
  const name &dac_id
);
	No issues.	No

ACTION
nominatecane(
  const name &cand,
  const eosio::asset &requestedpay,
  const name &dac_id
);
	No issues.	Yes

ACTION
withdrawcane(
  const name &cand,
  const name &dac_id
);
	No issues.
	Wrapper for `removeCandidate`
		Yes

ACTION
firecand(
  const name &cand,
  const bool lockupStake,
  const name &dac_id
);
	No issues.
	Wrapper for `removeCandidate`
		Yes

ACTION
resigncust(
  const name &cust,
  const name &dac_id
);
	No issues.
	Wrapper for `removeCandidate`
		Yes

ACTION
firecust(
  const name &cust,
  const name &dac_id
);
	No issues.
	Wrapper for `removeCustodian`.
		Yes

ACTION
appointcust(
  const vector<name> &cust,
  const name &dac_id
);
	No issues.	No

ACTION
updatebio(
  const name &cand,
  const std::string &bio,
  const name &dac_id
);
	No issues. Action does not persist any data.	No

[[eosio::action]] inline void
stprofile(
  const name &cand,
  const std::string &profile,
  const name &dac_id
) {
  require_auth(cand);
};
	No issues. Function does nothing.	Yes

[[eosio::action]] inline void	No issues. Function does nothing.	No

stprofileuns(
  const name &cand,
  const std::string &profile
) {
  require_auth(cand);
};		

ACTION
updatereqpay(
  const name &cand,
  const eosio::asset &requestedpay,
  const name &dac_id
);
	No issues.	Yes

ACTION
votecust(
  const name &voter,
  const std::vector<name> &newvotes,
  const name &dac_id
);
	No issues.	Yes

ACTION
voteproxy(
  const name &voter,
  const name &proxy,
  const name &dac_id
);
	No issues.	Yes

ACTION regproxy(const name &proxy, const name &dac_id);	No issues.	No
ACTION unregproxy(const name &proxy, const name &dac_id);	No issues.	No
ACTION newperiod(const std::string &message, const name &dac_id);	No issues.	Yes
ACTION runnewperiod(const std::string &message, const name &dac_id);	No issues. Remark: newperiod_components.cpp#267 very unlikely divide by zero possibility. You could add `check(token_current_supply > 0);`	No (But same as above)
ACTION claimpay(const uint64_t payid, const name &dac_id);	No issues.	Yes
ACTION removecuspay(const uint64_t payid, const name &dac_id);	No delta.	No
ACTION rejectcuspay(const uint64_t payid, const name &dac_id);	No delta.	Yes
ACTION paycpu(const name &dac_id);	No delta.	No
resetvotes (DEBUG ONLY)	No delta.	
resetcands (DEBUG ONLY)	No delta.	

ACTION
setperm(
  const name &cand,
  const name &permission,
  const name &dac_id
);
	No delta.	No

Helper Functions	Notes
void updateVoteWeight(name custodian, int64_t weight, name internal_dac_id);	No delta.
void updateVoteWeights(const vector<name> &votes, int64_t vote_weight, name internal_dac_id);	No delta.
int64_t get_vote_weight(name voter, name dac_id);	No issues.
void modifyVoteWeights(int64_t vote_weight, vector<name> oldVotes, vector<name> newVotes, name internal_dac_id);	No issues.
void modifyProxiesWeight(int64_t vote_weight, name oldProxy, name newProxy, name dac_id);	No delta.
void assertPeriodTime(contr_config &configs, contr_state &currentState);	No issues.
void distributeMeanPay(name internal_dac_id);	No issues.
vector<eosiosystem::permission_level_weight> get_perm_level_weights(const custodians_table &custodians, const name &dac_id);	No issues.
void add_all_auths(const name &accountToChange, const vector<eosiosystem::permission_level_weight> &weights, const name &dac_id, const bool msig = false);	No issues.
void add_all_auths_msig(const name &accountToChange, vector<eosiosystem::permission_level_weight> &weights, const name &dac_id);	Remark: Function implementation not found.
void add_auth_to_account(const name &accountToChange, const uint8_t threshold, const name &permission, const name &parent, vector<eosiosystem::permission_level_weight> weights, const bool msig = false);	No issues.
void setMsigAuths(name dac_id);	No issues.
void setCustodianAuths(name internal_dac_id);	No issues.
void transferCustodianBudget(const dacdir::dac &dac);	No issues.
void removeCustodian(name cust, name internal_dac_id);	No delta.
void removeCandidate(name cust, bool lockupStake, name internal_dac_id);	No delta.
void allocateCustodians(bool early_election, name internal_dac_id);	No issues.
bool permissionExists(name account, name permission);	No delta.
bool _check_transaction_authorization(const char *trx_data, uint32_t trx_size, const char *pubkeys_data, uint32_t pubkeys_size, const char *perms_data, uint32_t perms_size);	No delta.
permission_level getCandidatePermission(name account, name internal_dac_id);	No delta.
void validateUnstake(name code, name cand, name dac_id);	Remark: Function implementation not found.
void validateUnstakeAmount(const name &code, const name &cand, const asset &unstake_amount, const name &dac_id);	No delta.
void validateMinStake(name account, name dac_id);	No delta.

Test Case		Notes
updateconfige	Should fail for a dac_id without a dac	No issues.
	Should fail for invalid high auth threshold	No issues.
	Should fail for invalid mid auth threshold	No issues.
	Should fail for invalid low auth threshold	No issues.
	Should fail for invalid num elected	No issues.
	Should fail for invalid max votes	No issues.
	Should fail for invalid period length	No issues.
	Should fail for invalid initial quorum percent	No issues.
	Should fail for invalid quorum percent	No issues.
	Should succeed with valid params	No issues.
nominatecane	With a staking enabled DAC - with unregistered member - should fail with error	No issues.
	With a staking enabled DAC - with registered member - with insufficient staked funds - should fail with error	No issues.
	With a staking enabled DAC - with registered member - with sufficient staked funds - should succeed	No issues.
	With a staking disabled DAC - with unregistered member - should fail with error	No issues.
	With a staking disabled DAC - with registered member - should succeed	No issues.
	With a staking disabled DAC - with registered member - should fail to unstake	No issues.
	With a staking disabled DAC - with registered member - should fail to unstake zero amount	No issues.
votecust	with no votes - candidates should have 0 for total_votes	No issues.
	with no votes - state should have 0 the total_weight_of_votes	No issues.
	After voting - votes table should have rows	No issues.
	After voting - only candidates with votes have total_votes values	No issues.
	After voting - state should have increased the total_weight_of_votes	No issues.
	vote values after transfers - assert preconditions for vote values for custodians	No issues.
	vote values after transfers - assert preconditions for total vote values on state	No issues.
	vote values after transfers - after transfer to non-voter values should reduce for candidates and total values	No issues.
	vote values after transfers - total vote values on state should have changed	No issues.
voteproxy	After voting but before proxy voting - votes table should have rows	No issues.

	After voting but before proxy voting - only candidates with votes have total_votes values	No issues.
	After voting but before proxy voting - state should have increased the total_weight_of_votes	No issues.
	Before registering as a proxy - voteproxy should fail with not registered error	No issues.
	Registering as proxy - without correct auth - should fail with auth error	No issues.
	Registering as proxy - with correct auth - should succeed	No issues.
	After proxy voting - votes table should have rows	No issues.
	After proxy voting - only candidates with votes have total_votes values	No issues.
	After proxy voting - state should have increased the total_weight_of_votes	No issues.
	After proxy voting - vote values after transfers - assert preconditions for vote values for custodians	No issues.
	After proxy voting - vote values after transfers - assert preconditions for total vote values on state	No issues.
	After proxy voting - vote values after transfers - after transfer to non-voter values should reduce for candidates and total values	No issues.
	After proxy voting - vote values after transfers - total vote values on state should have changed	No issues.
	After proxy voting - after unregproxy - with wrong auth - should fail	No issues.
	After proxy voting - after unregproxy - with correct auth - should succeed	No issues.
	After proxy voting - with non proxy member - should fail	No issues.
	After proxy voting - values of votes after unregproxy should be updated.	No issues.
	After proxy voting - should reduce vote weight for existing votes	No issues.
	After proxy voting - total vote values on state should have changed	No issues.
regproxy / unregproxy	without an activation account - before a dac has commenced periods - without enough INITIAL candidate value voting - should fail with voter engagement too low error	No issues.
	without an activation account - before a dac has commenced periods - with enough INITIAL candidate value voting - without enough candidates with > 0 votes to fill the configs - should fail with not enough candidates error	No issues.
	without an activation account - before a dac has commenced periods - with enough INITIAL candidate value voting - with enough candidates to fill the configs - should succeed with custodians populated	No issues.
	without an activation account - before a dac has commenced periods - with enough INITIAL candidate value voting - with enough candidates to fill the configs - Should have highest ranked votes in custodians	No issues.
	without an activation account - before a dac has commenced periods - with enough INITIAL candidate value voting - with enough candidates to fill the configs - Custodians should not yet be paid	No issues.

	without an activation account - before a dac has commenced periods - with enough INITIAL candidate value voting - with enough candidates to fill the configs - should set the auths	No issues.
	Calling newperiod before the next period is due - should fail with too calling newperiod too early error	No issues.
	Calling new period after the period time has expired - should succeed	No issues.
	Calling new period after the period time has expired - custodians should have been paid	No issues.
	Calling new period after the period time has expired - custodians should the mean pay from the valid requested pays. (Requested pay exceeding the max pay should be ignored from the mean.)	No issues.
	Calling new period after the period time has expired - claimpay should fail without receiver's authority	No issues.
	Calling new period after the period time has expired - claimpay should transfer the money	No issues.
newperiod	without an activation account - before a dac has commenced periods - should fail with voter engagement too low error	No issues.
	without an activation account - before a dac has commenced periods - with enough INITIAL candidate value voting - without enough candidates with > 0 votes to fill the configs - should fail with not enough candidates error	No issues.
	without an activation account - before a dac has commenced periods - with enough INITIAL candidate value voting - with enough candidates to fill the configs - custodians should have been paid	No issues.
	without an activation account - before a dac has commenced periods - with enough INITIAL candidate value voting - with enough candidates to fill the configs - custodians should the mean pay from the valid requested pays. (Requested pay exceeding the max pay should be ignored from the mean.)	No issues.
	without an activation account - before a dac has commenced periods - with enough INITIAL candidate value voting - with enough candidates to fill the configs - claimpay should fail without receiver's authority	No issues.
	without an activation account - before a dac has commenced periods - with enough INITIAL candidate value voting - with enough candidates to fill the configs - claimpay should transfer the money	No issues.
	Dac with 0 payment for custodians - new period should succeed	No issues.
	Dac with 0 payment for custodians - custodians should not have been paid	No issues.
resigncust	should fail with incorrect auth returning auth error	No issues.
	With correct auth - for a currently elected custodian - without enough elected candidates to replace - should fail with not enough candidates error	No issues.
	With correct auth - for a currently elected custodian - with enough elected candidates to replace a removed candidate - should succeed with lockup of stake	No issues.
	With correct auth - for an unelected candidate - should fail with not current custodian error	No issues.
withdrawcane	should fail for unregistered candidate with not current candidate error	No issues.

	should fail with incorrect auth returning auth error	No issues.
	with correct auth - for a currently elected custodian - should succeed with lockup of stake active from previous election	No issues.
	with correct auth - for an unelected candidate - should succeed	No issues.
	with correct auth - for an unelected candidate - should allow unstaking without a timelock error	No issues.
firecand	should fail for unregistered candidate with not current candidate error	No issues.
	should fail with incorrect auth returning auth error	No issues.
	with correct auth - for a currently elected custodian - should succeed with lockup of stake active from previous election	No issues.
	with correct auth - for an unelected candidate should succeed	No issues.
firecust	should fail with incorrect auth returning auth error	No issues.
	with correct auth - for a currently elected custodian - should succeed with lockup of stake	No issues.
	With correct auth - for an unelected candidate - should fail with not current custodian error	No issues.
stakeobsv	with candidate in a registered candidate locked time - with less than the locked up quantity staked - should fail to unstake	
appointcust	should fail without correct auth	No issues.
	should succeed with correct auth	No issues.
	should fail with existing custodians appointed	No issues.

Test Coverage

The amount of actions tested divided by the number of total actions.

13 / 25 = 52%

No methodology definitively proves the absence of vulnerabilities. Following assessment and remediation, modifications to an application, its platform, network environment, and new threat vectors may result in new application security vulnerabilities.