// HALBORN

Mars Protocol - Outposts Incentives

CosmWasm Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **September 26th, 2022 - October 14th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY 3
CONTACTS 3
1 EXECUTIVE OVERVIEW 5
1.1 INTRODUCTION 6
1.2 AUDIT SUMMARY 6
1.3 TEST APPROACH & METHODOLOGY 7
RISK METHODOLOGY 7
1.4 SCOPE 9
2 ASSESSMENT SUMMARY & FINDINGS OVERVIEW 10
3 FINDINGS & TECH DETAILS 11
3.1 (HAL-01) PRIVILEGED ADDRESS CAN BE TRANSFERRED WITHOUT CONFIRMATION - LOW 13
Description 13
Code Location 13
Risk Level 14
Recommendation 14
Remediation Plan 14
3.2 (HAL-02) MISSING VALIDATION ROUTINE FOR PARAMETER RESPONSIBLE FOR EMISSIONS - LOW 15
Description 15
Code Location 15
Risk Level 17
Recommendation 17
Remediation plan 17
3.3 (HAL-03) POTENTIAL RISKY FUNCTIONS - INFORMATIONAL 18

Description 18

Code Location 18

Risk Level 19

Recommendation 19

Remediation plan 19

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	09/26/2022	Jakub Heba
0.2	Draft Version	09/30/2022	Jakub Heba
0.3	Draft Review	09/30/2022	Gabi Urrutia
1.0	Remediation Plan	10/07/2022	Jakub Heba
1.1	Remediation Plan Review	10/10/2022	Gabi Urrutia
1.2	Document Update	10/14/2022	Jakub Heba
1.3	Document Update Review	10/17/2022	Gabi Urrutia

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com

Luis Quispe Gonzales	Halborn	Luis.QuispeGonzales@halborn.com
Jakub Heba	Halborn	Jakub.Heba@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Mars Protocol engaged Halborn to conduct a security audit on their smart contracts beginning on September 26th and ending on October 14th. The security assessment was scoped to the smart contracts provided in the GitHub repository Outposts, commit hashes and further details can be found in the Scope section of this report.

1.2 AUDIT SUMMARY

The team at Halborn assigned a full-time security engineer to audit the security of the smart contract. The security engineer is a blockchain and smart-contract security expert with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit is to:

- Ensure that smart contract functions operate as intended
- Identify potential security issues with the smart contracts

In summary, Halborn identified some security risks that were partially addressed by the Mars Protocol team:

- Add a safe transfer ownership logic, preferably in a form of a two-steps process.
- Enhance security of the contract by implementing validation routine when setting asset incentive.
- Remove risky functions from the contract if there is no strong argumentation in the documentation for using such functionality.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of this audit. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of the Rust code and can quickly identify items that do not follow security best practices. The following phases and associated tools were used throughout the term of the audit:

- Research into architecture and purpose.
- Smart contract manual code review and walkthrough.
- Manual testing by custom scripts and fuzzers.
- Scanning of Rust files for vulnerabilities, security hotspots or bugs.
- Static Analysis of security for scoped contract, and imported functions.
- Testnet deployment.

RISK METHODOLOGY:

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

RISK SCALE - LIKELIHOOD

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.
3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.

1 - Very unlikely issue will cause an incident.

RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.
1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL

10 - CRITICAL
9 - 8 - HIGH
7 - 6 - MEDIUM
5 - 4 - LOW
3 - 1 - VERY LOW AND INFORMATIONAL

1.4 SCOPE

First round of testing (Sep 26th - Sep 30th):

1. CosmWasm Smart Contracts
    (a) Repository: outposts
    (b) Commit ID: e9fbc50dec55f68964cf33da0f4051c0cf3d6202
    (c) Contracts in scope:
        i. incentives
    (d) Packages in scope:
        i. outpost

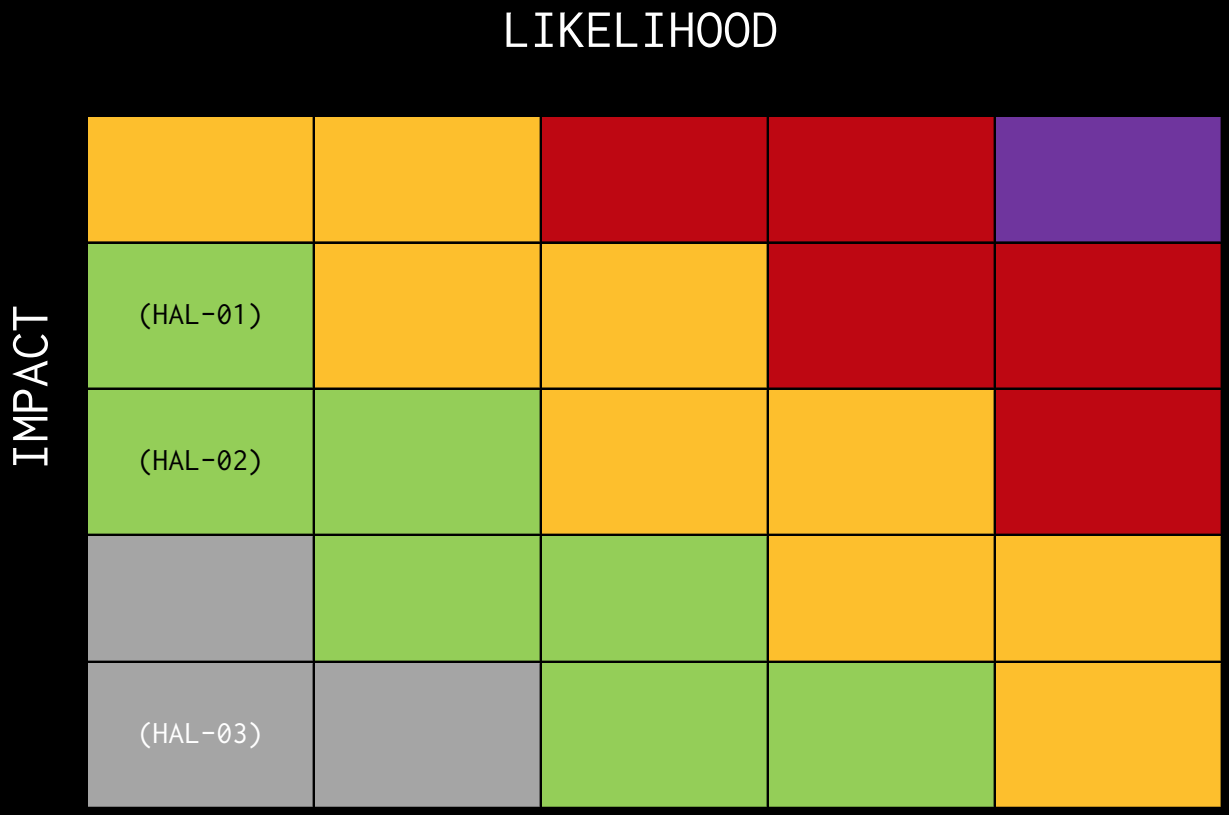
Second round of testing (Oct 13th - Oct 14th):

1. CosmWasm Smart Contracts
    (a) Repository: outposts
    (b) Commit ID: e476501a784c78de1b7f350722febe6d77d3a35d
    (c) Contracts in scope:
        i. incentives
    (d) Packages in scope:
        i. outpost

Out-of-scope: External libraries and financial related attacks

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	0	2	1



SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
(HAL-01) PRIVILEGED ADDRESS CAN BE TRANSFERRED WITHOUT CONFIRMATION	Low	RISK ACCEPTED
(HAL-02) MISSING VALIDATION ROUTINE FOR PARAMETER RESPONSIBLE FOR EMISSIONS	Low	RISK ACCEPTED
(HAL-03) POTENTIAL RISKY FUNCTIONS	Informational	SOLVED - 10/05/2022

FINDINGS & TECH DETAILS

3.1 (HAL-01) PRIVILEGED ADDRESS CAN BE TRANSFERRED WITHOUT CONFIRMATION - LOW

Description:

An incorrect use of the execute_update_config function from the **incentives** contract could set the owner to an invalid address, unwillingly losing control of the contract, which cannot be undone in any way. Currently, the OWNER of the contracts can change its address using the aforementioned function in a single transaction and without confirmation from the new address.

Code Location:

Listing 1: contracts/incentives/src/contract.rs (Lines 286,291)

```
272 pub fn execute_update_config(
273     deps: DepsMut,
274     _env: Env,
275     info: MessageInfo,
276     owner: Option<String>,
277     address_provider: Option<String>,
278     mars_denom: Option<String>,
279 ) -> Result<Response, MarsError> {
280     let mut config = CONFIG.load(deps.storage)?;
281
282     if info.sender != config.owner {
283         return Err(MarsError::Unauthorized {});
284     };
285
286     config.owner = option_string_to_addr(deps.api, owner, config.
 ↳ owner)?;
287     config.address_provider =
288         option_string_to_addr(deps.api, address_provider, config.
 ↳ address_provider)?;
289     config.mars_denom = mars_denom.unwrap_or(config.mars_denom);
290
291     CONFIG.save(deps.storage, &config)?;
```

```
292
293     let response = Response::new().add_attribute("action", "
 ↳ outposts/incentives/update_config");
294
295     Ok(response)
296 }
```

Risk Level:

Likelihood - 1
Impact - 4

Recommendation:

The execute_update_config function should follow a two steps process, being split into set_owner and accept_owner functions. The latter one requiring the transfer to be completed by the recipient, effectively protecting the contract against potential typing errors compared to single-step OWNER transfer mechanisms.

Remediation Plan:

RISK ACCEPTED: The Mars team accepted the risk of this finding.

3.2 (HAL-02) MISSING VALIDATION ROUTINE FOR PARAMETER RESPONSIBLE FOR EMISSIONS - LOW

Description:

The emission_per_second parameter is not validated against a maximum or minimum acceptable value when being set or changed. In case when this value is set to very low value like 0, token holders will not receive any emissions, and in the case of a very high value, this may lead to a situation in which the contract owner may maliciously withdraw the funds from the contract.

Code Location:

Fragment of execute_set_asset_incentive:

Listing 2: contracts/incentives/src/contract.rs (Lines 92,124,129,135)

```
87 pub fn execute_set_asset_incentive(
88     deps: DepsMut,
89     env: Env,
90     info: MessageInfo,
91     denom: String,
92     emission_per_second: Uint128,
93 ) -> Result<Response, ContractError> {
94     // only owner can call this
95     let config = CONFIG.load(deps.storage)?;
96     let owner = config.owner;
97     if info.sender != owner {
98         return Err(MarsError::Unauthorized {}.into());
99     }
100
101     let red_bank_addr = address_provider::helpers::query_address(
102         deps.as_ref(),
103         &config.address_provider,
104         MarsContract::RedBank,
105     )?;
106
```

```
107     let new_asset_incentive = match ASSET_INCENTIVES.may_load(deps
 ↳ .storage, &denom)? {
108         Some(mut asset_incentive) => {
109             let market: red_bank::Market = deps.querier.
 ↳ query_wasm_smart(
110                 &red_bank_addr,
111                 &red_bank::QueryMsg::Market {
112                     denom: denom.clone(),
113                 },
114             )?;
115 
116             // Update index up to now
117             asset_incentive_update_index(
118                 &mut asset_incentive,
119                 market.collateral_total_scaled,
120                 env.block.time.seconds(),
121             )?;
122 
123             // Set new emission
124             asset_incentive.emission_per_second =
 ↳ emission_per_second;
125 
126             asset_incentive
127         }
128         None => AssetIncentive {
129             emission_per_second,
130             index: Decimal::zero(),
131             last_updated: env.block.time.seconds(),
132         },
133     };
134 
135     ASSET_INCENTIVES.save(deps.storage, &denom, &
 ↳ new_asset_incentive)?;
136 
137     let response = Response::new().add_attributes(vec![
138         attr("action", "outposts/incentives/set_asset_incentive"),
139         attr("denom", denom),
140         attr("emission_per_second", emission_per_second),
141     ]);
142     Ok(response)
143 }
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

It is recommended to add a validation routine inside execute_set_asset_incentive functions to ensure that bands for that value are properly enforced.

Remediation plan:

RISK ACCEPTED: The Mars team accepted the risk of this finding.

3.3 (HAL-03) POTENTIAL RISKY FUNCTIONS - INFORMATIONAL

Description:

The execute_execute_cosmos_msg allow executing CosmosMsg on behalf of the contract. However, the execute_execute_cosmos_msg is an administrative operation, thus the likelihood that it will be exploited via such an attack vector is extremely low. However, if someone chooses to do so, any kind of CosmosMsg will be executed on behalf of the contract, e.g: Withdraw tokens from red_bank. This exposes protocol to unnecessary risk.

Code Location:

Listing 3: contracts/incentives/src/contract.rs

```
298 pub fn execute_execute_cosmos_msg(
299     deps: DepsMut,
300     _env: Env,
301     info: MessageInfo,
302     msg: CosmosMsg,
303 ) -> Result<Response, MarsError> {
304     let config = CONFIG.load(deps.storage)?;
305
306     if info.sender != config.owner {
307         return Err(MarsError::Unauthorized {});
308     }
309
310     let response = Response::new()
311         .add_attribute("action", "outposts/incentives/
 ↳ execute_cosmos_msg")
312         .add_message(msg);
313
314     Ok(response)
315 }
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

If there is no strong argumentation in the documentation for using such functionality, removing such methods from the contract is recommended to lower the risk of exposition.

Remediation plan:

SOLVED: The issue was solved in commit df0a6bc952f1cae9c57703d61b3572bce8324b57.

THANK YOU FOR CHOOSING

// HALBORN