Mars Protocol - Outposts address-provider

CosmWasm Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **September 16th, 2022 - October 28th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	09/20/2022	Thiago Mathias
0.2	Document Update	09/20/2022	Thiago Mathias
0.3	Draft Review	09/25/2022	Timur Guvenkaya
0.4	Draft Review	09/26/2022	Gabi Urrutia
1.0	Remediation Plan	09/30/2022	Thiago Mathias
1.1	Remediation Plan Review	09/30/2022	Gabi Urrutia
1.2	Document Update	10/13/2022	Thiago Mathias
1.3	Document Review	10/18/2022	Gabi Urrutia
1.4	Document Update	10/28/2022	Elena Maranon
1.5	Document Update Review	11/02/2022	Gabi Urrutia
1.6	Remediation Plan	11/04/2022	Elena Maranon
1.6	Remediation Plan Review	11/09/2022	Elena Maranon

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
Thiago Mathias	Halborn	Thiago.Mathias@halborn.com
Elena Maranon	Halborn	Elena.Maranon@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Mars Protocol engaged Halborn to conduct a security audit on their smart contracts beginning on September 16th, 2022 and ending on October 28th, 2022. The security assessment was scoped to the smart contracts provided to the Halborn team.

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 found the contract to follow secure development best practices, resulting in only a low finding with negligible security impact.

1.3 TEST APPROACH & METHODOLOGY

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 16th - Sep 23rd):

1. CosmWasm Smart Contracts
   - (a) Repository: outposts
   - (b) Commit ID: 191ac5e5bf655d02331df6ff04f9756caced989d
   - (c) Contracts in scope:
     - i. address-provider
   - (d) Packages in scope:
     - i. outpost

Second round of testing (Oct 11th - Oct 13th):

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

Third round of testing (Oct 26th - Oct 28th):

1. CosmWasm Smart Contracts
   - (a) Repository: outposts
   - (b) Commit ID: dc909b26a353b7353c0017a2fdc85794ebf2276d

(c) Contracts in scope:

   i. address-provider

(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	1	1

LIKELIHOOD

IMPACT

🟨	🟨	🟥	🟥	🟪
🟩 (HAL-01)	🟨	🟨	🟥	🟥
🟩	🟩	🟨	🟨	🟥
⬜	🟩	🟩	🟨	🟨
⬜ (HAL-02)	⬜	🟩	🟩	🟨

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
(HAL-01) PRIVILEGED ADDRESS CAN BE TRANSFERRED WITHOUT CONFIRMATION	Low	RISK ACCEPTED
(HAL-02) OUTDATED SCHEMA	Informational	ACKNOWLEDGED

FINDINGS & TECH DETAILS

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

Description:

An incorrect use of the transfer_ownership function from the **address-provider** 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/address-provider/src/contract.rs

```rust
pub fn transfer_ownership(
    deps: DepsMut,
    sender: Addr,
    new_owner: String,
) -> Result<Response, ContractError> {
    let mut config = CONFIG.load(deps.storage)?;

    assert_owner(&sender, &config.owner)?;
    assert_valid_addr(deps.api, &new_owner, &config.prefix)?;

    config.owner = new_owner.clone();
    CONFIG.save(deps.storage, &config)?;

    Ok(Response::new()
        .add_attribute("action", "outposts/address-provider/transfer_ownership")
        .add_attribute("previous_owner", sender)
        .add_attribute("new_owner", new_owner))
}
```

Risk Level:

Likelihood - 1
Impact - 4

Recommendation:

The transfer_ownership 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 Protocol team accepted the risk of this finding.

3.2 (HAL-02) OUTDATED SCHEMA - INFORMATIONAL

Description:

After the last commit of the code, the JSON Schema has got outdated, and it still contains references to MarsContract definition. Consequently, the auto-generated typescript files from scripts folder also contains wrong references.

Code Location:

Listing 2: Resources affected

```
schema/mars-address-provider/mars-address-provider.json
scripts/types/generated/mars-address-provider/MarsAddressProvider.client.ts
scripts/types/generated/mars-address-provider/MarsAddressProvider.react-query.ts
scripts/types/generated/mars-address-provider/MarsAddressProvider.types.ts
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to update the JSON Schema.

Remediation Plan:

ACKNOWLEDGED: The Mars Protocol team acknowledged this finding.

THANK YOU FOR CHOOSING

// HALBORN