Quantstamp

Andromeda Core (OS ADOs And ADOBase)

Executive Summary

This audit report was prepared by Quantstamp, the leader in blockchain security.

Type	CosmWasm-Based Smart Contract Development Framework
Timeline	2023-11-15 through 2024-01-30
Language	Rust
Methods	Architecture Review, Unit Testing, Functional Testing, Manual Review
Specification	developer guide
Source Code	• andromedaprotocol/andromeda-core #d320af1
Auditors	• Ed Zulkoski Senior Auditing Engineer • Faycal Lalidji Senior Auditing Engineer • Mostafa Yassin Auditing Engineer • Poming Lee Senior Auditing Engineer

Documentation quality	Medium
Test quality	Low
Total Findings	46 Unresolved: 6 Fixed: 34 Acknowledged: 5 Mitigated: 1
High severity findings	4 Fixed: 4
Medium severity findings	24 Unresolved: 4 Fixed: 18 Acknowledged: 2
Low severity findings	8 Unresolved: 1 Fixed: 7
Undetermined severity findings	6 Unresolved: 1 Fixed: 3 Acknowledged: 2
Informational findings	4 Fixed: 2 Acknowledged: 1 Mitigated: 1

Summary of Findings

This report outlines the security audit conducted by Quantstamp on Andromeda, a comprehensive smart contract development framework. Andromeda is designed to facilitate the creation of DApps on cosmos-based blockchains, offering both a no-code, UI-driven approach and a more traditional module/library option for developers. The framework also includes a set of cross-cosmos-chain deployed contracts for developer interaction with these contracts across various chains.

During our audit, Quantstamp identified over 44 issues. A significant concern is the project's testing quality, particularly the lack of tests for negative paths and more complex scenarios; most tests focus solely on "happy paths". This limitation has led to doubts about the project's overall security and readiness for production. The audit team is not confident that all potential issues have been identified.

Key issues identified include:

- Certain code paths are prone to consistent failure.
- Flaws in validation and checks functions, undermining the effectiveness of these safeguards.
- Several edge cases in VFS ADO could lead to a Denial of Service.
- Exploitable design elements in the VFS ADO and `AMPPkt` relaying mechanisms.
- Inadequate access controls in multiple components, potentially allowing attacks.

We recommend a second, comprehensive audit following the resolution of all reported issues and substantial enhancements in code and testing procedures.

2024-03-05 Update: In the latest fix-check process, Quantstamp discovered two additional issues within the code. Nonetheless, overall the majority of issues classified as high to medium severity have now been addressed or formally recognized. The development team is committed to resolving the outstanding issues in future updates and intends to conduct additional rounds of audits in the future.

ID	DESCRIPTION	SEVERITY	STATUS
AND-1	**Incorrect IBC Path Declaration in `create_cross_chain_message()` Leading to Cross-Chain Functionality Breakdown**	High	Fixed
AND-2	**Missing Access Control for the `permission.rs`**	High	Fixed
AND-3	**DoS by Depleting an ADO's Balance at Economics ADO**	High	Fixed
AND-4	**Incorrect implementation in `validate_andr_addresses()`**	High	Fixed
AND-5	**`register_user()` with a Long Username Would Stop the System From Serving the Created User.**	Medium	Fixed
AND-6	**Inaccurate Coin Merging Due to Duplicate Handling in `merge_coins()`**	Medium	Fixed
AND-7	**Incomplete Funds Handling in `handle_local()`**	Medium	Fixed
AND-8	**Action Fees Incorrectly Associated with ADO Version Instead of Type**	Medium	Fixed
AND-9	**Paths Misinterpretation in `resolve_home_path()`**	Medium	Unresolved
AND-10	**`handle_add_app_component()` Fails to Enforce 50 Component Maximum Limit**	Medium	Fixed
AND-11	**`handle_local()` Will Panic if the Funds Sent Are `None` and the Message Is Not `Binary::default`**	Medium	Fixed
AND-12	**Potential Channel to Chain Mismatch when a Updating `channelInfo` for a Chain**	Medium	Fixed
AND-13	**Privileged Roles and Ownership**	Medium	Acknowledged
AND-14	**Publishing ADOs at Scale Can Introduce Malicious ADOs**	Medium	Acknowledged
AND-15	**Malicious Modules Cannot Be Removed if They Are Immutable**	Medium	Unresolved
AND-16	**Missing Input Validation Can Cause the Cross-Chain Flow to Panic**	Medium	Fixed
AND-17	**No Validation Is Done on the Fee Amount for `update_action_fees`**	Medium	Unresolved
AND-18	**`local_path_to_vfs_path()` Does Not Consider the User Path**	Medium	Fixed
AND-19	**Storing `ado_version` Instead of an Ado's Type in**	Medium	Fixed

ID	DESCRIPTION	SEVERITY	STATUS
	`store_code_id()`		
AND-20	**Incorrect Branching in** `handle_ibc()`	Medium	Fixed
AND-21	**No Max Limit in** `get_subdir()` **May Lead to DoS**	Medium	Fixed
AND-22	**Splitting on** `":"` **in an** `asset_string` **May Lead to Incorrect Payment Denominations**	Medium	Fixed
AND-23	**Unrestricted Access in VFS function** `add_parent_path()`	Medium	Fixed
AND-24	**Flaw in Handling Symlink Components in Andromeda App Instantiation**	Medium	Fixed
AND-25	**The Function** `handle_ibc_direct()` **Will Always Fail**	Medium	Fixed
AND-26	**ADO Developors Not Able to Upgrade Their ADO if** `version` **Is Incorrectly Specified as** `"latest"` **the First Time**	Medium	Fixed
AND-27	**The Symlink Can Be Nested and Could Lead to DoS**	Medium	Fixed
AND-28	**Contract Upgrades Allow for Spoofing Legitimate Apps/ADOs**	Medium	Unresolved
AND-29	**Lack of Two-Step Verification in Ownership Transfer**	Low	Fixed
AND-30	**Flawed User Registration Logic in** `contract.rs:register_user()` **Allowing Multiple Usernames per Address**	Low	Fixed
AND-31	**Condition in** `register_user()` **Should Trigger Return, Not Continue Logic Execution**	Low	Fixed
AND-32	**Inconsistency in** `PATH_REGEX` **: Failure to Recognize Local Paths Starting with** `"./"`	Low	Fixed
AND-33	`ado_type` **May Be Set to the Empty String**	Low	Fixed
AND-34	`andromeda-economics` **Does Not Have** `reply()`	Low	Fixed
AND-35	`validate_component_name()` **Accepts** `.` **as a Component Name**	Low	Fixed
AND-36	**There Is No Predefined Pause Mechanism in the System for Emergency Handling**	Low	Unresolved
AND-37	**Unnecessary Tilde Symbol Check in** `resolve_lib_path()` **Function**	Informational	Fixed
AND-38	**Risk of Arithmetic Overflows in Cosmwasm Projects Compiled in Release Mode without** `overflow-checks`	Informational	Mitigated
AND-39	**Unresolved TODOs**	Informational	Acknowledged

ID	DESCRIPTION	SEVERITY	STATUS
AND-40	**Inconsistent Access Control for Undocumented Helper Functions in** `ADOContract`	• **Informational**	Fixed
AND-41	`register_user_cross_chain()` **Permits Assignment of a Single Username to Multiple Addresses Across Different Blockchains**	• **Undetermined**	Acknowledged
AND-42	**Usernames May Be Valid CosmWasm Addresses Not Equal to** `info.sender`	• **Undetermined**	Fixed
AND-43	**The** `amp_receive()` **in Kernel Allows Any Smart Contract to Call It**	• **Undetermined**	Fixed
AND-44	**Any sender can get their** `AMPMsg` **relayed by** `kernel`	• **Undetermined**	Acknowledged
AND-45	**The Possibility of Nesting App and App Components in VFS Path**	• **Undetermined**	Unresolved
AND-46	**The** `match` **Case in** `get_asset_string()` **Is Incorrect**	• **Undetermined**	Fixed

Assessment Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

> **Disclaimer**
> Only features that are contained within the repositories at the commit hashes specified on the front page of the report are within the scope of the audit and fix review. All features added in future revisions of the code are excluded from consideration in this report.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

1. Code review that includes the following
    1. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
    2. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
    3. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:

    1. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    2. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarity, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Scope

1. Quantstamp was commissioned to audit a specific set of files, as listed below. Any files not explicitly mentioned are deemed outside the scope of this audit.
2. This audit is based on the assumption that anything other than the listed files, such as the front-end, the used libraries, Cosmos, Cosmwasm, the Inter Blockchain Communication (IBC) protocol, among other existing infrastructures, work correctly, safely, and as intended.
3. Only features that are contained within the repositories at the commit hash specified on the front page of the report are within the scope of the audit and fix review. All features added in future revisions of the code are excluded from consideration in this report.

Files Included

- contracts\app\andromeda-app-contract\src\lib.rs
- contracts\app\andromeda-app-contract\src\state.rs
- contracts\app\andromeda-app-contract\src\contract.rs
- contracts\os\andromeda-adodb\src\lib.rs
- contracts\os\andromeda-adodb\src\state.rs
- contracts\os\andromeda-adodb\src\contract.rs
- contracts\os\andromeda-vfs\src\lib.rs
- contracts\os\andromeda-vfs\src\state.rs
- contracts\os\andromeda-vfs\src\contract.rs
- contracts\os\andromeda-economics\src\lib.rs
- contracts\os\andromeda-economics\src\state.rs
- contracts\os\andromeda-economics\src\contract.rs
- contracts\os\andromeda-kernel\src\ack.rs
- contracts\os\andromeda-kernel\src\ibc.rs
- contracts\os\andromeda-kernel\src\sudo.rs
- contracts\os\andromeda-kernel\src\execute.rs
- contracts\os\andromeda-kernel\src\lib.rs
- contracts\os\andromeda-kernel\src\query.rs
- contracts\os\andromeda-kernel\src\proto.rs
- contracts\os\andromeda-kernel\src\state.rs
- contracts\os\andromeda-kernel\src\contract.rs
- contracts\os\andromeda-kernel\src\reply.rs
- packages\std\macros\src\lib.rs
- packages\std\src\error.rs
- packages\std\src\ado_contract\execute.rs
- packages\std\src\ado_contract\withdraw.rs
- packages\std\src\ado_contract\query.rs
- packages\std\src\ado_contract\ownership.rs
- packages\std\src\ado_contract\mod.rs
- packages\std\src\ado_contract\permissioning.rs
- packages\std\src\ado_contract\state.rs
- packages\std\src\ado_contract\app.rs
- packages\std\src\ado_contract\modules\execute.rs
- packages\std\src\ado_contract\modules\query.rs
- packages\std\src\ado_contract\modules\mod.rs
- packages\std\src\ado_contract\instantiate.rs
- packages\std\src\lib.rs
- packages\std\src\common\response.rs
- packages\std\src\common\expiration.rs
- packages\std\src\common\withdraw.rs
- packages\std\src\common\mod.rs
- packages\std\src\common\rates.rs
- packages\std\src\common\queries.rs
- packages\std\src\common\context.rs
- packages\std\src\amp\recipient.rs
- packages\std\src\amp\mod.rs
- packages\std\src\amp\addresses.rs
- packages\std\src\amp\messages.rs

- packages\std\src\os\adodb.rs
- packages\std\src\os\kernel.rs
- packages\std\src\os\aos_querier.rs
- packages\std\src\os\mod.rs
- packages\std\src\os\economics.rs
- packages\std\src\os\vfs.rs
- packages\std\src\ado_base\ado_type.rs
- packages\std\src\ado_base\version.rs
- packages\std\src\ado_base\withdraw.rs
- packages\std\src\ado_base\ownership.rs
- packages\std\src\ado_base\hooks.rs
- packages\std\src\ado_base\block_height.rs
- packages\std\src\ado_base\mod.rs
- packages\std\src\ado_base\permissioning.rs
- packages\std\src\ado_base\modules.rs
- packages\std\src\ado_base\operators.rs
- packages\std\src\ado_base\kernel_address.rs

Findings

AND-1
Incorrect IBC Path Declaration in `create_cross_chain_message()` Leading to Cross-Chain Functionality Breakdown

• High Fixed

File(s) affected: `contracts/app/andromeda-app-contract/src/execute.rs`

Description: The function `create_cross_chain_message()` has an issue in the way it forms the IBC path for cross-chain components. Specifically, the current logic uses an incorrect IBC path format `"ibc://{curr_chain}/home/{owner}/{app_name}/{name}"` instead of the correct `"ibc://{chain}/home/{owner}/{app_name}/{name}"`. The code snippet is presented below:

```
if chain == chain_info.chain_name {
    AppComponent {
        name,
        ado_type: component.ado_type,
        component_type: ComponentType::New(instantiate_msg),
    }
} else {
    AppComponent {
        name: name.clone(),
        ado_type: component.ado_type,
        component_type: ComponentType::Symlink(AndrAddr::from_string(format!(
            "ibc://{curr_chain}/home/{owner}/{app_name}/{name}"
        ))),
    }
}
```

In this snippet, when the `chain` variable does not match `chain_info.chain_name`, a symlink component is incorrectly created with the IBC path using the current chain (`curr_chain`) instead of `chain` where the component is deployed. This error can disrupt the application's cross-chain logic, affecting functionality and leading to potential issues in inter-chain interactions.

Recommendation: To resolve this, modify the `else` clause to correctly utilize the `chain` variable in the IBC path, ensuring that the symlink component accurately points to the resource on the intended chain.

AND-2 Missing Access Control for the `permission.rs`

• High Fixed

File(s) affected: `packages/std/src/ado_contract/permission.rs`

Description: All functions in `permission.rs` are not implementing access control properly. A function will call `is_contract_owner()`, which will either return true or false. However, the return value is never used. This can allow an attacker to give themselves whitelist permission on any action they want on the application.

```
Self::is_contract_owner(self, ctx.deps.storage, ctx.info.sender.as_str())?;
```

Here is a PoC:

```
fn test_user_can_add_malicious_permission() {
    let mut deps = mock_dependencies_custom(&[]);
    let env = mock_env();
    let info = mock_info("creator", &[]);
    let inst_msg = InstantiateMsg {
        app_components: vec![],
        name: String::from("Some App"),
        owner: Some("owner".to_string()),
        kernel_address: MOCK_KERNEL_CONTRACT.to_string(),
        chain_info: None,
    };
    instantiate(deps.as_mut(), env.clone(), info.clone(), inst_msg).unwrap();

    // * sending the msg as "attacker"
    let info = mock_info("attacker", &[]);
    let action = "action";
    let actor = "attacker";

    // * -------------- PART 1 Setting Whitelist Permission For any user * -------------- * //
    // * preparing the AndromedaMsg
    let msg_data = AndromedaMsg::SetPermission {
        actor: AndrAddr::from_string((actor)),
        action: action.to_string(),
        permission: Permission::Whitelisted(None),
    };
    let exploit_msg = to_binary(&msg_data).unwrap();
    let amp_message = AMPMsg::new(env.clone().contract.address, exploit_msg, None);
    let packet = AMPPkt::new("attacker", "attacker", vec![amp_message]);
    let msg = ExecuteMsg::AMPReceive(packet);
    let res = execute(deps.as_mut(), env.clone(), info, msg);
    assert!(res.is_ok());
    // * reading attackers's permission
    let query_msg = QueryMsg::Permissions { actor: "attacker".to_string(), limit: None,
start_after: None };
    let res = query(deps.as_ref(), env.clone(), query_msg).unwrap();
    let permissions: Vec<PermissionInfo> = from_binary(&res).unwrap();
    let permission = &permissions[0];
    assert_eq!(permission.permission, Permission::Whitelisted(None));
    assert_eq!(permission.actor, "attacker".to_string());
    assert_eq!(permission.action, "action".to_string());
    // * -------------------------- PART 1 ends* ---------------------------------- * //

    // * from here, the user can either bypass an aciton that already exist (by setting user's
permission to whitelist(None))
    // * or they can pass the is_permissioned function for action that does not exist on the
permissioned_actions mapping
    // * there is also no access control on any of the permissions function, it just checks if
the caller is
    // * the contract owner, but doesn't do anything with the return value.
    // * now, we show that any user can add any action to the permissioned actions

     // * ------------------ PART 2 Adding Malicious Actions * --------------------- * //
    let info = mock_info("attacker", &[]);
    let msg_data = AndromedaMsg::PermissionAction { action: "malicious_action".to_string() };
    let exploit_msg = to_binary(&msg_data).unwrap();
    let amp_message = AMPMsg::new(env.clone().contract.address, exploit_msg, None);
    let packet = AMPPkt::new("attacker", "attacker", vec![amp_message]);
    let msg = ExecuteMsg::AMPReceive(packet);
    let res = execute(deps.as_mut(), env.clone(), info, msg);
    assert!(res.is_ok());
    // * reading permissions
    let query_msg = QueryMsg::PermissionedActions { };
    let res = query(deps.as_ref(), env.clone(), query_msg).unwrap();
    let action: Vec<String> = from_binary(&res).unwrap();
    assert_eq!(action[0],"malicious_action".to_string());
     // * -------------------------- PART 2 Ends* ------------------------------ * //
```

```
}
```

Recommendation: The ownership check should be wrapped in an `ensure` block.

```
ensure!(
    Self::is_contract_owner(self, ctx.deps.storage, ctx.info.sender.as_str())?,
    ContractError::Unauthorized {}
    );
```

AND-3
DoS by Depleting an ADO's Balance at Economics ADO

• High Fixed

> **Update**
> Fixed by disabled the functionality of having ADO or App pay the fee.

File(s) affected: `contracts/os/andromeda-economics/src/contract.rs`

Description: Based on the current implementation in `execute_pay_fee()` any end user of an ADO can deplete the target ADO's balance in Economics ADO by using the target ADO's relevant functions to indirectly call this function repeatedly.

Exploit Scenario:

1. User Bob wants to DoS an ADO called "ADO1".
2. User Bob know that calling "ADO1.funct1" would let ADO1 to invoke `execute_pay_fee()` .
3. User Bob write a keeper bot to monitor and repeatedly call "ADO1.funct1" until ADO" has 0 balance.
4. When ADO1's balance is emptied, all end users that have no balance in Economics ADO would not be able to use ADO1 anymore.
5. User Bob can use this exploit to simutaneously DoS all the ADOs deployed to DoS Andromeda.

Recommendation: Have the end users pay for the fees. The Andromeda team can encourage ADO devs to set fees to be 0 in the early stage of the project to attract end users if it is needed.

AND-4 Incorrect implementation in `validate_andr_addresses()`

• High Fixed

File(s) affected: `packages/std/src/ado_contract/execute.rs`

Description:

The returned boolean value of `address.is_addr(deps.api);` in `validate_andr_addresses()` is never checked. This violates the whole point of this function.

Recommendation: Return an `Err` immediately whenever `address.is_addr(deps.api) == false` .

AND-5
`register_user()` with a Long Username Would Stop the System From Serving the Created User.

• Medium Fixed

File(s) affected: `contracts/os/andromeda-vfs/src/execute.rs` , `packages/std/src/os/vfs.rs`

Description: Calling `execute.rs::register_user()` with a long username would stop the system from serving the created user. This is because of the pattern `([A-Za-z0-9\.\-_]{1,40}(/)?)` in `pub const PATH_REGEX: &str = r"^(([A-Za-z0-9]+://)?([A-Za-z0-9\.\-_]{1,40})?(/)?(home|lib)/)|(~(/)?))([A-Za-z0-9\.\-_]{1,40}(/)?)+$";` . All the operations related to the VFS path of all the users with username that is longer than 40 (note that it is unstricted in `pub const USERNAME_REGEX: &str = r"^[a-z0-9]+$";`) will not work.

Recommendation: Change `vfs.rs::USERNAME_REGEX` from `r"^[a-z0-9]+$"` into `r"^[a-z0-9]{1,40}$"` .

AND-6
Inaccurate Coin Merging Due to Duplicate Handling in `merge_coins()`

Medium | Fixed

File(s) affected: `packages/std/src/common/mod.rs`

Description: The `merge_coins()` function exhibits a flaw in how it processes duplicates within the `coins_to_add: Vec<Coin>`. Specifically, the function uses `coins_to_add.iter().find(|&c| c.denom == coin.denom)` to search for a `Coin` with the same denomination as the current `coin`. However, this approach only considers the first occurrence of a matching denomination and disregards any subsequent duplicates. Consequently, if `coins_to_add` contains multiple `Coin` instances with the same denomination, the function inaccurately merges only the amount of the first found instance, leading to potential errors in the total coin calculation.

A similar issue is observed in the `deduct_funds()` function, which assumes there is only one matching `Coin` in `coins` that corresponds to `funds.denom`.

Recommendation: Revise `merge_coins()` to aggregate all `Coin` instances with the same denomination in `coins_to_add`. Similarly, revise `deduct_funds()`.

AND-7 Incomplete Funds Handling in `handle_local()`

Medium | Fixed

File(s) affected: `contracts/os/andromeda-kernel/src/execute.rs`

Description: The function `handle_local()` first considers the case for funds transfers in the main if branch. The following line adds an attribute to the response:

```
.add_attributes(vec![
    attr(format!("recipient:{sequence}"), recipient_addr),
    attr(format!("bank_send_amount:{sequence}"), funds[0].to_string()),
]);
```

If there are multiple coin transfers in `funds`, the message will only report the first item. This could affect external listeners relying upon the attributes.

Further, in the main `else` branch (corresponding to a non-default message), we have the following which only considers `funds[0]`:

```
let sub_msg = new_packet.to_sub_msg(
    recipient_addr.clone(),
    Some(vec![funds[0].clone()]),
    ReplyId::AMPMsg.repr(),
)?;
```

Recommendation: Iterate through `funds` to add an attribute for each item. Ensure that all funds are accounted for when creating a sub-message.

AND-8
Action Fees Incorrectly Associated with ADO Version Instead of Type

Medium | Fixed

File(s) affected: `contracts/os/andromeda-adodb/src/contract.rs`

Description: Based on the client's communication, the `update_action_fees()` function should exclusively utilize the ADO type. However, there's an inconsistency in `andromeda-adodb/src/contract.rs:publish()`, where it invokes `update_action_fees()` using the version instead of the type. This deviation needs to be addressed to align with the specified requirements of using only the ADO type.

Recommendation: We recommend resolving this issue by utilizing the ADO type rather than the ADO version.

AND-9 Paths Misinterpretation in `resolve_home_path()`

Medium | Unresolved

> **Alert**
>
> The example paths mentioned above are now resolvable; however, there remains an edge case, `~/app1`, that cannot currently be resolved correctly.

File(s) affected: `contracts/os/andromeda-vfs/src/state.rs`

Description: The `resolve_home_path()` function exhibits an issue in processing certain path patterns. Paths like `"~username1/appname"` or `"~username1/username2"` (possible malicious path) are deemed valid by regex validation. However, due to flawed logic, the function incorrectly resolves the `username_or_address` to `"appname"` or `"username2"` instead of the intended `username1`. This error stems from the implementation of `resolve_home_path()`, particularly in how it interprets the intended username. The problematic code is as follows:

```
let username_or_address = if parts[0].starts_with('~') && parts.len() == 1 {
    parts[0].remove(0);
    parts[0].as_str()
} else {
    parts[1].as_str()
};
```

For a path such as `~username1/appname`, `parts.len()` equals two, leading to the execution of the else branch. This results in `username_or_address` being incorrectly set to `appname`.

Recommendation: To address this issue update the `resolve_home_path()` function to correctly interpret and resolve user directory paths. Ensure that the function accurately identifies the primary username in complex path structures and handles them appropriately.

AND-10

`handle_add_app_component()` Fails to Enforce 50 Component Maximum Limit

Medium Fixed

File(s) affected: `contracts/app/andromeda-app-contract/src/execute.rs`, `contracts/app/andromeda-app-contract/src/reply.rs`, `contracts/app/andromeda-app-contract/src/contract.rs`, `contracts/app/andromeda-app-contract/src/state.rs`

Description: As outlined in the Andromeda protocol documentation, there is a stipulated hard limit of 50 components for an app. This limit is enforced during the instantiation of an app. However, it appears that this restriction is not applied in the `handle_add_app_component()` function. When an app owner adds a new component using a direct component addition message, the function does not check against the 50 component maximum limit. This oversight can lead to excessive gas consumption, as several functions depend on the number of components and iterate over the components array.

Currently, the absence of a maximum limit check in the system could lead to additional complications, as detailed below:

Issue Name: Unexpected Outcomes When Adding Over 100 App Components to an Application

Issue Details: The process of adding a new app component is initiated in the following function:

```
pub fn add_app_component(
    storage: &mut dyn Storage,
    component: &AppComponent,
) -> Result<u64, ContractError> {
    let idx = ADO_IDX.may_load(storage)?.unwrap_or(1u64); // Retrieve the "next" index of
components in the app
    ADO_DESCRIPTORS.save(storage, &idx.to_string(), component)?; // Map `message id ->
AppComponent`
    ADO_IDX.save(storage, &(idx + 1))?;

    Ok(idx)
}
```

The system then processes the `SubMsg` response in the `reply()` function. The `_` branch in the `match ReplyId::from_repr(msg.id)` suggests that `msg.id` serves as the index in `ADO_DESCRIPTORS`, linking an ID to its corresponding `AppComponent`. However, if `msg.id` exceeds 100, it conflicts with the predefined `ReplyId` enum, potentially causing the system to stop functioning without any notification.

The `ReplyId` enum is as follows:

```
#[EnumRepr(type = "u64")]
pub enum ReplyId {
    ClaimOwnership = 101,
    AssignApp = 102,
    RegisterPath = 103,
    CrossChainCreate = 104,
}
```

Recommendation: To resolve this issue, it is recommended to update the `handle_add_app_component()` to include a check that ensures the total number of components does not exceed the maximum limit of 50.

AND-11

`handle_local()` Will Panic if the Funds Sent Are `None` and the Message Is Not `Binary::default`

Medium Fixed

File(s) affected: `contracts/os/andromeda-kernel/execute.rs`

Description: The function `handle_local()` has two main branches. The first one is for if the message is a `Binary::default`, and the second one is for anything else.

A binary default message must have funds attached, and this check is handled by the contract. However, a nonbinary default message may not have funds attached, meaning that the funds array will be empty.

In the `else` branch, the `amp_msg` is generated as:

```
        let amp_msg = AMPMsg::new(
                    recipient_addr.clone(),
                    message.clone(),
                    Some(vec![funds[0].clone()]),
                );
```

Notice how the funds array is accessed, while there are no checks to see if it has a non-zero length. This will cause the execution to panic due to an out-of-bound exception.

Here is a PoC:

```
fn test_handle_local_can_panic() {
    let mut deps = mock_dependencies_custom(&[]);
    let info = mock_info("creator", &[]);
    let env = mock_env();
    instantiate(
        deps.as_mut(),
        env.clone(),
        info.clone(),
        InstantiateMsg {
            owner: None,
            chain_name: "andromeda".to_string(),
        },
    )
    .unwrap();

    let msg = ExecuteMsg::AssignChannels {
        ics20_channel_id: None,
        direct_channel_id: None,
        chain: "chain2".to_string(),
        kernel_address: Addr::unchecked("kernal2").to_string(),
    };

    let amp_msg = AMPMsg::new(Addr::unchecked("john"), to_binary(&msg).unwrap(), None);
    let amp_packet = AMPPkt::new(Addr::unchecked("john"),Addr::unchecked("john"),vec![amp_msg]);
    let exec_msg = ExecuteMsg::AMPReceive((amp_packet));

    execute(deps.as_mut(), env.clone(), info.clone(), exec_msg).unwrap(); // <---- panics at
'index out of bounds:
```

```
                                                                the len is 0 but the
index is 0
}
```

Notice how the fund array is passed as `None` in `AMPMsg::new`. Indeed, the execution panics at the expected location.

Recommendation: It is recommended to first check if the funds being sent are `Some` or it is nonempty before accessing the array. An example could be:

```
if funds.is_empty() {
        f = None;
    } else {
        f = Some(vec![funds[0].clone()]);
    }
```

AND-12

Potential Channel to Chain Mismatch when a Updating `channelInfo` for a Chain

Medium Fixed

File(s) affected: `contracts/os/andromeda-kernel-execute.rs`

Description: The function `assign_channels` updates a mapping of a chain to a `channelInfo` struct that contains information for IBC cross-chain interaction. However, in case of a `channelInfo` being already set for a given `chain`, and the owner calls the function again for the same `chain`, it is possible to have a mismatch between the mappings `CHAIN_TO_CHANNEL` and `CHANNEL_TO_CHAIN`.

For example:

Suppose the `admin` sets the `channelInfo` for `chain2` with `ics20_channel_id = ics20` and `direct_channel_id = direct`. This will properly set up the mappings in both `CHAIN_TO_CHANNEL` and `CHANNEL_TO_CHAIN`.

However, if the admin then wants to update the `channelInfo` for `chain2`, say `direct = new_direct`, and say `ics20 = None`. This will not update the `CHANNEL_TO_CHAIN` for the `ics20` channel, and it will continue to point to `chain2`. That is because the write to `CHANNEL_TO_CHAIN` is not invoked if the channel passed is `None`:

```
    if let Some(channel) = channel_info.ics20_channel_id.clone() {
        CHANNEL_TO_CHAIN.save(execute_env.deps.storage, &channel, &chain)?;
     }
```

Below is a PoC that shows this example:

```
fn test_chain_to_channel_mismatch() {
    let mut deps = mock_dependencies_custom(&[]);
    let info = mock_info("creator", &[]);
    let env = mock_env();
    instantiate(
        deps.as_mut(),
        env.clone(),
        info.clone(),
        InstantiateMsg {
            owner: None,
            chain_name: "andromeda".to_string(),
        },
    )
    .unwrap();

    let msg = ExecuteMsg::AssignChannels {
        ics20_channel_id: Some("ics20".to_string()),
        direct_channel_id: Some("direct".to_string()),
        chain: "chain2".to_string(),
        kernel_address: Addr::unchecked("kernal2").to_string(),
    };

    execute(deps.as_mut(), env.clone(), info.clone(), msg).unwrap();
```

```
    let query_msg: andromeda_std::os::kernel::QueryMsg =
andromeda_std::os::kernel::QueryMsg::ChannelInfo { chain: "chain2".to_string() };
    let res:andromeda_std::os::kernel::ChannelInfoResponse = from_binary(&query(deps.as_ref(),
env.clone(), query_msg).unwrap()).unwrap();
    println!("{:?}", res);
    // ChannelInfoResponse { ics20: Some("ics20"), direct: Some("direct"), kernel_address:
"kernal2", supported_modules: [] }

    let msg = ExecuteMsg::AssignChannels {
        ics20_channel_id: None,
        direct_channel_id: None,
        chain: "chain2".to_string(),
        kernel_address: Addr::unchecked("kernal2").to_string(),
    };

    execute(deps.as_mut(), env.clone(), info.clone(), msg).unwrap();

    let query_msg: andromeda_std::os::kernel::QueryMsg =
andromeda_std::os::kernel::QueryMsg::ChannelInfo { chain: "chain2".to_string() };
    let res:andromeda_std::os::kernel::ChannelInfoResponse = from_binary(&query(deps.as_ref(),
env.clone(), query_msg).unwrap()).unwrap();
    println!("{:?}", res);
    // ChannelInfoResponse { ics20: None, direct: None, kernel_address: "kernal2",
supported_modules: [] }

    let channel = CHANNEL_TO_CHAIN.load(deps.as_mut().storage, "ics20").unwrap();
    println!("channel ics20 still maps to {:?}", channel);
    // Channel ics20 still maps to "chain2"

}
```

To be more specific, the issue with `CHANNEL_TO_CHAIN` not being updated, and more specifically that the original channel (`ics20` or `direct` above) are never removed (regardless of if we pass in new values).

Here is another example:

Suppose our initial call includes parameters `<chain2, initial_ics20_channel, initial_direct_channel>` , and we call again with `<chain2, other_ics20_channel, other_direct_channel>` . In the code below,

```
if let Some(channel) = channel_info.direct_channel_id.clone() {
    CHANNEL_TO_CHAIN.save(execute_env.deps.storage, &channel, &chain)?;
}
if let Some(channel) = channel_info.ics20_channel_id.clone() {
    CHANNEL_TO_CHAIN.save(execute_env.deps.storage, &channel, &chain)?;
}
```

it adds new entries to `CHANNEL_TO_CHAIN` without removing the old ones, that is, `CHANNEL_TO_CHAIN` now has all four entries `<initial_ics20_channel, chain2>` , `<initial_direct_channel, chain2>` , `<other_ics20_channel, chain2>` , and `<other_direct_channel, chain2>` .

This may allow continued use of a channel that the Andromeda team wants to remove.

Recommendation: It is recommended to overwrite the `CHANNEL_TO_CHAIN` mapping regardless of the channel values; so no mismatch happens.

AND-13 Privileged Roles and Ownership

Medium Acknowledged

> **Update**
> The dev stated that "Control remains centralised. Possible (and likely) shift to DAO management structure in future.".

File(s) affected: `contracts/os/andromeda-adodb/execute.rs`

Description: The `andromeda-adodb` allows the owner to update the publisher and the fees for a given ADO. This can also be done by any address that holds the operator role. This kind of design is centralized and gives a lot of power to the protocol owners.

The following functions are used to perform these updates:

1. `execute_update_publisher()`
2. `execute_update_action_fees()`

Recommendation: It is recommended to outline this centralization concern in user-facing documentation.

AND-14
Publishing ADOs at Scale Can Introduce Malicious ADOs

Medium | Acknowledged

> **Update**
> The dev stated that they would add it in the future.

Description: The protocol plans to allow for third parties to create ADO components, which will then be published by the protocol and will be available for integration by various users.

At scale, this poses a security risk of not being able to audit and verify the security of all the new ADO components.

Recommendation: Currently, the development team lacks an "ADO Assessment Process" for evaluating new ADOs. It is advisable to implement a comprehensive security protocol for scrutinizing each new ADO, thereby minimizing potential vulnerabilities. This protocol could encompass a multi-step process: firstly, conducting KYC/KYB checks on the publishers of new ADOs; followed by an internal review within Andromeda, focusing on the code and its interactions with other ADOs; and finally, securing an external audit, etc.

AND-15
Malicious Modules Cannot Be Removed if They Are Immutable

Medium | Unresolved

File(s) affected: `packages/std/src/ado_contract/modules/execute.rs` , `packages/std/src/ado_contract/modules/mod.rs`

Description: The `ado_base` allows for the registering of modules. In the future, third parties will be allowed to create modules that will be available to users of Andromeda.

Modules hold the `is_mutable` field, which is used to tell the `ado_base` contract whether this module can be updated later on or not (through being removed or replaced).

In case a user registers a module, then later discovers that it is vulnerable or incompatible with their application, and given that the property `is_mutable = false` , they will not be able to remove the module or replace it.

Recommendation: It is recommended to review this design and give the option for owners of ADO applications to force remove certain third-party modules if they cause issues.

It is possible to only allow immutability for modules that are deemed to be used by the Andromeda protocol.

AND-16
Missing Input Validation Can Cause the Cross-Chain Flow to Panic

Medium | Fixed

> **Update**
> The edge case is handled gracefully now.

Description: The function `assign_channels()` is callable by the contract owner and allows the registering of `channel_info` on different chains. This is the data structure used:

```
let channel_info = ChannelInfo {
    ics20_channel_id,
    direct_channel_id,
    kernel_address,
```

```
            supported_modules: vec![],
        };
```

The parameters `ics20_channel_id` and `direct_channel_id` are options that could be `None` or `Some`, however, this is not checked before being saved in the `CHAIN_TO_CHANNEL` mapping.

This means that during sending cross-chain messages, `direct_channel_id` or `ics20_channel_id` may be set as `None`, and when `unwrap()` is called on them, this will cause execution to panic.

The following is a PoC that shows this case:

```
   fn test_assign_channel_panics() {
        let mut deps = mock_dependencies_custom(&[]);
        let info = mock_info("creator", &[]);
        let env = mock_env();
        instantiate(
            deps.as_mut(),
            env.clone(),
            info.clone(),
            InstantiateMsg {
                owner: None,
                chain_name: "andromeda".to_string(),
            },
        )
        .unwrap();
        let msg = ExecuteMsg::AssignChannels {
            ics20_channel_id: None,
            direct_channel_id: None,
            chain: "chain2".to_string(),
            kernel_address: Addr::unchecked("kernal2").to_string(),
        };
        execute(deps.as_mut(), env.clone(), info.clone(), msg).unwrap();

        let internal_msg = InternalMsg::RegisterUserCrossChain {
            username: "name".to_string(),
            address: Addr::unchecked("addr").to_string(),
            chain: "chain2".to_string(),
        };

        let msg = ExecuteMsg::Internal(internal_msg);

       execute(deps.as_mut(), env.clone(), info.clone(), msg); // <---- panics
    }
```

Recommendation: Perform checks in `assign_channels()` to make sure that `direct_channel_id` and `ics20_channel_id` are passed as `Some`.

Another alternative is to use `unwrap_or()` or `unwrap_or_default()` to throw specific errors and use a pre-defined default value.

AND-17
No Validation Is Done on the Fee Amount for

Medium Unresolved

`update_action_fees`

> **Update**
> Added asset string check. However, there is currently no check for the fee amount yet.

File(s) affected: `contracts/os/andromeda-adodb/execute.rs`

Description: The `update_action_fees()` allows the owner to update the fees for a given action on a given ADO. However, the fees are not validated in any way, and they seem to be a flat amount rather than a percentage. This can cause some flows to revert if the fee required is greater than the available balance.

Also, the `update_action_fees()` in ADODB does not check the string format of `ActionFee.asset`. If the string is incorrectly formatted, `execute_pay_fee()` in Economics ADO would not be able to correctly resolve it at `let asset =`

`asset_string.split(':').last().unwrap()` and all the transactions related to it would be reverted.

Recommendation: It is recommended to do validation for the fee amounts and to have an upper bound for them if they are flat. Or have them as a percentage with defined boundaries. Furthermore, add relevant checks to `update_action_fees()`, such as, native token denom check, cw20 token address check, and the check of the existence of the `:` char.

AND-18 `local_path_to_vfs_path()` Does Not Consider the User Path

Medium Fixed

File(s) affected: `packages/std/src/amp/addresses.rs`

Description: `get_raw_address_from_vfs()` is used in multiple places to convert an `AndrAddr` into raw adderess. Inside this function, `local_path_to_vfs_path()` is used to convert paths from starting with "./" into starting with "/home/{app_contract}/".

Recommendation: Replace "/home/{app_contract}/" by "/home/{username}/{app_contract}/" where `username` stands for the username of the specific instantiator of the app.

AND-19 Storing `ado_version` Instead of an Ado's Type in `store_code_id()`

Medium Fixed

> **Update**
> This issue was fixed in `32218bb`. However, the dev team mentioned "We're reverting the ADODB's mapping between Code ID and type to also include the version then adjusting our queries to use get_type from the version. It's a substantial improvement for frontend UX I'm told.". This means that the exact code that is going to be used for production is going to be different from what Quantstamp audited during the fix-check.

File(s) affected: `contracts/os/andromeda-adodb/src/state.rs`, `contracts/os/andromeda-adodb/src/contract.rs`

Description: Currently `state::store_code_id()` stores `ado_version` to `ADO_TYPE` instead of storing the type of the ADO. This would result in `curr_type.unwrap() == ado_version.get_type()` being always false and thus block this code path. This incorrectness is expected to also have effects on `contract::query_action_fee()` and `contract::query_action_fee_by_code_id()`.

Recommendation: Save `ado_version.get_type()` instead of `ado_version`.

AND-20 Incorrect Branching in `handle_ibc()`

Medium Fixed

File(s) affected: `contracts/os/andromeda-kernel/src/execute.rs`

Description:

Based on the implemented code logic in `handle_ibc_hooks()` and `handle_ibc_direct()`, the condition `!self.message().funds.is_empty()` of the `if` statement in `handle_ibc()` is incorrect. It will call `handle_ibc_hooks()` to handle non-funds transfer request and `handle_ibc_direct()` for funds transfer request. This will cause all the IBC messages to be reverted and not sent.

Recommendation: Use `self.message().funds.is_empty()` instead of `!self.message().funds.is_empty()`.

AND-21 No Max Limit in `get_subdir()` May Lead to DoS

Medium Fixed

File(s) affected: `contracts/os/andromeda-vfs/src/state.rs`

Description: In `get_subdir()`, there is no range max set when checking all items in a given directory. This could have potential gas issues if any other contract relies on the `subdir()` query, and may even impact users simply wishing to query their own directory's contents.

Note that a separate issue discusses the possibility of users polluting the home directories of other users, which would allow griefing attacks on `get_subdir()`.

Recommendation: Allow specifying a range max limit for `get_subdir()`.

AND-22
Splitting on `":"` in an `asset_string` May Lead to Incorrect Payment Denominations

• Medium ⓘ Fixed

File(s) affected: `contracts/os/andromeda-economics/src/contract.rs`

Description: In `execute_pay_fee()`, we get the asset by taking the `asset_string and splitting on ":"`, and only considering the string after the last `":"`. This could be problematic in systems which may utilize different tokens with similar names (possibly maliciously).

Exploit Scenario: Suppose the `ActionFee = { asset: "scam:coin", ...}`, where `scam:coin` is some low priced token, but the `coin` part is the name of a legitimate and valuable token also utilized by the `ADO`. If the user were to perform an action that has this fee (thinking they are spending `scam:coin`), they would be tricked into losing their `coin` balance instead.

Recommendation: If the string split operation is necessary, ensure that the full `asset_string` value cannot also be a valid coin.

AND-23 Unrestricted Access in VFS function `add_parent_path()`

• Medium ⓘ Fixed

File(s) affected: `contracts/os/andromeda-vfs/src/execute.rs`

Description: The Virtual File System (VFS) is currently exposed to a security risk due to the absence of access control in its `add_parent_path()` function. This gap allows a malicious actor to deploy a contract and utilize `add_parent_path()` to inappropriately link a harmful contract to a parent path of a renowned project. For instance, an attacker could alter message parameters to mimic a trusted address, e.g., Uniswap, by setting the `parent_address` in the `VFSExecuteMsg::AddParentPath` and sending the `AddParentPath` message from the malicious contract:

```
VFSExecuteMsg::AddParentPath {
    name: "routerUpgrade",
    parent_address: AndrAddr::from_string("/home/uniswap/"),
};
```

The core of this issue lies in the lack of validation for the `parent_address` input, leading to potential misassociations of parent paths with an attacker's contract.

Recommendation: We advise restricting the ability to add a child to the `parent_address` exclusively to its owner. Alternatively, the execution of `add_parent_path()` could be limited when the `parent_address` is specified, allowing its use solely by a caller that is both recognized and verified as a trusted address.

AND-24
Flaw in Handling Symlink Components in Andromeda App Instantiation

• Medium ⓘ Fixed

File(s) affected: `contracts/app/andromeda-app-contract/src/contract.rs`

Description: In the Andromeda application, there is an issue in the `contract.rs:instantiate()` function regarding the handling of symlink components. The function fails to properly enforce that `InstantiateMsg.app_components` should exclude any symlink components when the `InstantiateMsg.chain_info` vector is not null or empty, except if this is the intended design and symlink need to be passed as valid component when instantiating a new Andromeda application. This oversight is significant because the `execute.rs:create_cross_chain_message()` function is designed to modify the symlink chain to align with the current chain, subsequently propagating it across all chains specified in `InstantiateMsg.chain_info`. Please refer to the code below for the default case in the match expression (symlink included):

```
_ => AppComponent {
    name: name.clone(),
    ado_type: component.ado_type,
    component_type: ComponentType::Symlink(AndrAddr::from_string(format!(
        "ibc://{curr_chain}/home/{owner}/{app_name}/{name}"
    ))),
},
```

The current implementation could lead to unintended behavior or security vulnerabilities in the cross-chain messaging process.

Recommendation: Amend the `instantiate()` function to include a validation check that explicitly prohibits the inclusion of symlink components in `InstantiateMsg.app_components` whenever `InstantiateMsg.chain_info` is not null or not empty.

AND-25 The Function `handle_ibc_direct()` Will Always Fail

Medium Fixed

> **Update**
> Added `!` before `Binary::default().eq(message)`.

File(s) affected: `contracts/os/kernel/execute.rs`

Description: The function `handle_ibc_direct()` is only invoked when there are no funds attached to the message, however, it performs the following check:

```
ensure!(
        Binary::default().eq(message),
        ContractError::InvalidPacket {
            error: Some("Cannot send an empty message without funds via IBC".to_string())
        }
    );
```

By definition, a Binary message is a send message, which must have funds attached, so this check will never pass, since when `handle_ibc_direct()` is invoked, no funds are attached.

Here is a PoC:

```
fn test_handle_ibc_fails() {
    let mut deps = mock_dependencies_custom(&[]);
    let info = mock_info("user", &[]);
    let env = mock_env();
    let chain = "andromeda";
    instantiate(
        deps.as_mut(),
        env.clone(),
        info.clone(),
        InstantiateMsg {
            owner: None,
            chain_name: "andromeda".to_string(),
        },
    )
    .unwrap();
    let channel_info = ChannelInfo {
        kernel_address: MOCK_FAKE_KERNEL_CONTRACT.to_string(),
        ics20_channel_id: Some("1".to_string()),
        direct_channel_id: Some("2".to_string()),
        supported_modules: vec![],
    };
    KERNEL_ADDRESSES
        .save(
            deps.as_mut().storage,
            VFS_KEY,
            &Addr::unchecked(MOCK_VFS_CONTRACT),
        )
        .unwrap();
    CHAIN_TO_CHANNEL
        .save(deps.as_mut().storage, chain, &channel_info)
        .unwrap();
    let dummy_msg = ExecuteMsg::UpsertKeyAddress {
        key: "key".to_string(),
        value: "value".to_string(),
    };
    let amp_msg = AMPMsg::new("ibc://andromeda/..", to_binary(&dummy_msg).unwrap(), None);
    let packet = AMPPkt::new("user", "user", vec![amp_msg]);
    let msg = ExecuteMsg::AMPReceive((packet));
    let res = execute(deps.as_mut(), env, info, msg);
    // * message fails even though it is a non-default binary message
```

```
        assert!(res.is_err());
    }
```

Recommendation: Revise the check to make sure it works as expected

AND-26
ADO Developors Not Able to Upgrade Their ADO if `version` Is Incorrectly Specified as `"latest"` the First Time

Medium Fixed

> **Update**
> It is redesigned.

File(s) affected: `contracts/os/andromeda-adodb/src/contract.rs`

Description:

In `publish()`, for the first publish, `version` can be set as any string (such as `"latest"`, `"1"`, `"2"`, `"3"`) and could even not be a `semver::Version`. If it is set as `"latest"`, it would cause the problem of the ADO not being able to upgrade later because of being blocked here: `let current_version = semver::Version::parse(&ado_version.0).unwrap();`.

Recommendation: This check

```
    ensure!(
            version.validate(),
            ContractError::InvalidADOVersion { msg: None }
        );
```

should also validate if the given `version` matches a `semver::Version` string.

AND-27 The Symlink Can Be Nested and Could Lead to DoS

Medium Fixed

File(s) affected: `contracts/os/andromeda-vfs/src/execute.rs`

Description:

The `add_symlink()` allows nested symlinks. Thus edge cases like these can happen and would cause DoS when it happens.

1. symlink1 ⇒ symlink2 ⇒ (...) ⇒ symlinkN (long links that would reach the computational resource and/or gas limit of resolving it).
2. symlink1 ⇒ symlink2 ⇒ symlink1 (looped links).

Recommendation: Implement a mechanism in `add_symlink()` to ensure that 1) no loops and 2) the number of nested symlink is limited. Think of more edge cases and prevent them from happening.

AND-28
Contract Upgrades Allow for Spoofing Legitimate Apps/ADOs

Medium Unresolved

File(s) affected: `All`

Description: (This issue is found in commit hash: `e1a8000` during the fix-check stage of the audit.)

If the end-user deploys an app and is set as the `admin` during instantiation, they are currently able to migrate their instance to arbitrary code. This may lead to the following flow:

1. The malicious user deploys an arbitrary `app-contract` composed of off-the-shelf ADOs. These are considered trusted pieces of code vetted by the Andromeda team, and these associated components are registered in the ADODB and VFS.
2. They then invoke `migrate` on any of the components in their app, changing the contracts to arbitrary malicious code. This does not seem to perform any verification or update in the ADODB or VFS. In essence, this allows for users to have malicious code integrated into the Andromeda ecosystem while simultaneously spoofing legitimate ADOs. For example, the user might create an app with a staking component, but then upgrade that component to include a malicious backdoor to steal funds.

Recommendation: Restrict the migrate code ID to be limited to those in the ADODB.

AND-29 Lack of Two-Step Verification in Ownership Transfer

Low Fixed

> **Update**
> It is fixed and checked on a seperate commit `4c948e6` but not yet merged at the moment the audit team conduct fix-check based on the request by the dev.

File(s) affected: `packages/std/ado_contract/ownership.rs`

Description: The `execute_update_owner()` function within `ownership.rs` currently transfers ownership to a new address directly, without implementing a two-step verification process. This approach does not allow the new owner to validate the transfer actively or confirm their readiness to assume ownership.

As a result, this could lead to unintended ownership transfers if the new address is incorrect or if the supposed new owner is not informed or prepared for this change. Such a situation poses a risk, especially if the ownership transfer grants significant control over the contract's operations.

Recommendation: To mitigate this risk and enhance the security and robustness of the ownership transfer process, revise the `execute_update_owner()` function to include a two-step process. Initially, the function should only propose a new owner, storing this information within the contract's state.

The actual transfer of ownership should only occur after the proposed new owner sends a message that executes a separate function to accept ownership.

AND-30 Flawed User Registration Logic in `contract.rs:register_user()` Allowing Multiple Usernames per Address

Low Fixed

File(s) affected: `contracts/os/andromeda-vfs/src/execute.rs`

Description: The `register_user()` function in `contract.rs` within the VFS exhibits problematic behavior, the function permits an Externally Owned Account (EOA) to call `register_user()` multiple times with different usernames. This allows multiple usernames to be associated with a single address.

The following code snippet seems to be at the origin of the issue and potentially problematic:

```
USERS.remove(env.deps.storage, username.as_str());
```

Following the code comment "Remove username registration from previous username," the `USERS.remove` operation in the code seems designed to unregister the previous username associated with the address. However, it incorrectly attempts to delete the current username. Moreover, this removal operation is redundant, as a subsequent save operation immediately rewrites the username that was just removed.

Recommendation: Ensure that `register_user()` properly updates the username of an address and deletes any references to old usernames.

AND-31 Condition in `register_user()` Should Trigger Return, Not Continue Logic Execution

Low Fixed

File(s) affected: `contracts/os/andromeda-vfs/src/execute.rs`

Description: The following code snippet in the `register_user()` function is intended to verify that the `sender` address matches the `current_user_address` :

```
if current_user_address.is_some() {
    ensure!(
        current_user_address.unwrap() == sender,
        ContractError::Unauthorized {}
    );
}
```

This condittion is designed to ensure that the `sender` is the same as the `current_user_address` . If this condition is met, the function should logically return early since all subsequent operations will not modify the contract state — the `sender` is already the

`current_user_address` and the username has been previously set.

Recommendation: We recommend modifying the function to return an appropriate message when the `sender` is equal to the `current_user_address`. This change will prevent unnecessary execution of the function's remaining code when the user is already registered, thereby enhancing efficiency and clarity.

AND-32
Inconsistency in `PATH_REGEX`: Failure to Recognize Local Paths Starting with `"./"`

• Low Fixed

File(s) affected: `packages/std/src/os/vfs.rs`

Description: The regex pattern defined in `PATH_REGEX` currently overlooks a case: it does not recognize local path names that start with `"./"`. This omission is inconsistent with the functionality in `is_vfs_path()`, where such paths are implemented and acknowledged as valid. The discrepancy arises when `validate_path_name()` uses `PATH_REGEX` for path validation, leading to potential misinterpretation or rejection of valid local paths starting with `"./"`.

Recommendation: Modify the `PATH_REGEX` to include local path names starting with `"./"`.

AND-33 `ado_type` May Be Set to the Empty String

• Low Fixed

File(s) affected: `contracts/os/andromeda-adodb/src/contract.rs`

Description: It is currently possible to publish an ADO with `ado_type == ""`. For example, if we have a call to `publish()` with `version = 1.2.3` and `ado_type = ""`, the corresponding `ADOVersion` is `@1.2.3`, which will pass the `validate()` check.

Recommendation: In `validate()`, ensure that element in slot `0` of the split on `@` is non-empty.

AND-34 `andromeda-economics` Does Not Have `reply()`

• Low Fixed

File(s) affected: `contracts/os/andromeda-economics/src/contract.rs`, `packages/std/src/ado_contract/execute.rs`

Description: There are some functions that set `SubMsg` as `ReplyOn.Error` whereas in `contracts/os/andromeda-economics/src/contract.rs` there is no `reply()` to handle them. They are:

1. `packages/std/src/ado_contract/execute.rs` on `L235`
2. `contracts/os/andromeda-economics/src/contract.rs` on `L349`.

Recommendation: Add a `reply()` to `packages/std/src/ado_contract/execute.rs` to handle the error properly.

AND-35
`validate_component_name()` Accepts `.` as a Component Name

• Low Fixed

File(s) affected: `packages/std/src/os/vfs.rs`

Description:

The `COMPONENT_NAME_REGEX` used in `validate_component_name()` accepts `.` as a component name. This is deemed to be unintended and could cause unexpected behavior of the system.

Recommendation: Modify `COMPONENT_NAME_REGEX` to exclude this edge case from happening.

AND-36
There Is No Predefined Pause Mechanism in the System for Emergency Handling

• Low Unresolved

Description: In smart contract design, incorporating a pause mechanism for emergency handling is essential. Given that Andromeda is poised to become a foundational infrastructure for numerous DeFi applications, potentially overseeing a substantial Total Value Locked (TVL), the inclusion of an emergency handling mechanism is critical. However, such a mechanism is currently absent.

Recommendation: We strongly recommend designing and implementing one. This includes exploring the feasibility and consequences of an 'escape hatch' to temporarily halt operations or, if necessary, forcefully remove ADOs during emergencies.

AND-37
Unnecessary Tilde Symbol Check in `resolve_lib_path()` Function

Informational | Fixed

File(s) affected: `contracts/os/andromeda-vfs/src/state.rs`

Description: It appears that the `resolve_lib_path()` function is mainly a duplicate of `resolve_home_path()`, containing a redundant check for the "home" directory, identifiable by the tilde symbol "~". This inclusion is erroneous as `resolve_lib_path()` is not designed to handle paths containing the tilde symbol. In practice, `resolve_pathname()` invokes `resolve_lib_path()` only for paths explicitly containing the `"lib"` segment, rendering the tilde-related check unnecessary.

Recommendation: The tilde symbol check in `resolve_lib_path()` is irrelevant and should be removed.

AND-38
Risk of Arithmetic Overflows in Cosmwasm Projects Compiled in Release Mode without `overflow-checks`

Informational | Mitigated

> **Update**
> The examples described in the issue have been fixed; however, there may be more widespread edge cases that require consideration.

File(s) affected: `packages/std/src/ado_base/permissioning.rs`

Description: When compiling a CosmWasm project to a WASM binary in release mode, arithmetic overflow checks are not included by default. They must be explicitly enabled. Without the `overflow-checks` flag set to `true` in the `Cargo.toml` file, arithmetic operations might overflow without triggering a panic in the program.

In the current project, the `overflow-checks` flag is correctly set to `true` in the root `Cargo.toml`. However, there's a possibility that this configuration might not be preserved in scenarios such as project modifications, forks, or copies.

Particular concern arises in functions like `consume_use()` located in `packages/std/src/ado_base/permissioning.rs`. In this function, the `uses` variable risks underflowing without triggering a panic, which could be exploited if overflow checks are not properly configured. This vulnerability underscores the importance of maintaining the overflow check logic.

Recommendation: We strongly encourage developers to implement logical overflow checks in their code by utilizing functions like `checked_add()`, `checked_sub()`, and similar safe arithmetic functions and return correct contract errors instead of panic.

AND-39 Unresolved TODOs

Informational | Acknowledged

> **Update**
> Some issues have been fixed, but there are still more TODOs remaining.

File(s) affected: `contracts/os/andromeda-adodb/src/contract.rs`, `contracts/os/andromeda-kernel/src/execute.rs`, `packages/std/src/amp/recipient.rs`

Description: Several TODO comments were identified across different modules and files in the codebase. These TODOs indicate areas that require further development, review, or optimization. Leaving TODOs unresolved in the code, especially in smart contracts, can lead to potential security vulnerabilities, incomplete features, or performance inefficiencies. This is raised as an issue due to the potential for these unresolved areas to contain critical bugs or oversights, especially in a smart contract environment where immutability and security are paramount.

Relevant code snippets:

1. `contracts/app/andromeda-app-contract/src/testing/mod.rs` : L335, L627
2. `contracts/os/andromeda-adodb/src/contract.rs` : L141
3. `contracts/os/andromeda-kernel/src/execute.rs` : L168
4. `packages/std/src/ado_contract/permissioning.rs` : L196, 223
5. `packages/std/src/amp/recipient.rs` : L67

Recommendation: Leaving TODOs unresolved can lead to a lack of clarity in the code's intention and functionality. It's important to address these issues to ensure the code is complete, secure, and functions as expected.

AND-40

Inconsistent Access Control for Undocumented Helper Functions in `ADOContract`

• Informational ⓘ Fixed

File(s) affected: `packages/std/src/ado_contract/withdraw.rs`

Description: There is inconsistent access control in the implemented functions, since `execute_withdraw()` contains an `is_owner_or_operator()` check, whereas this is not included for `add_withdrawable_token()` nor `remove_withdrawable_token()`. This may make it more error-prone for `ADOContract` creators when incorporating this functionality.

Recommendation: Add documentation for all helper/optional functions for `ADOContract`. Ensure that any assumptions that are required of the contract writer are explicitly noted. Where possible, ensure consistent access control policies amongst the implemented functions.

Note that this issue may extend beyond the `withdraw` functionality described above.

AND-41

`register_user_cross_chain()` Permits Assignment of a Single Username to Multiple Addresses Across Different Blockchains

• Undetermined ⓘ Acknowledged

File(s) affected: `/contracts/os/andromeda-vfs/src/execute.rs`

Description: The function `register_user_cross_chain()` permits any Externally Owned Account (EOA) that has already registered a username with its own address using `register_user()` to then employ `register_user_cross_chain()` for associating the same username with different addresses across multiple chains. This capability directly contradicts the constraints set by `register_user()`, which deliberately prevents such behavior (EOA can set a username for his own addressonly) to maintain username-address integrity.

The rationale behind allowing this in `register_user_cross_chain()` might be attributed to the fact that different blockchains can have distinct address formats and structures. Nonetheless, this feature presents a security concern. The ability to link a single username to well-known addresses on various chains can be exploited for social engineering attacks, misleading interactions, or impersonations.

Recommendation: To mitigate the risks and maintain the integrity of username assignments, it is advisable to implement additional controls in the `register_user_cross_chain()` function. Additionally, providing clear documentation of this behavior to end users is essential for transparency and informed usage.

AND-42

Usernames May Be Valid CosmWasm Addresses Not Equal to `info.sender`

• Undetermined ⓘ Fixed

File(s) affected: `contracts/os/andromeda-vfs/src/execute.rs`, `contracts/os/andromeda-vfs/src/state.rs`

Description: There is no constraint restricting a user from registering a valid CosmWasm address that is not equal to their own as their username. It is not clear why a user would consider this, however it may lead to attacks that attempt to spoof another user. Conversely, if an honest user chooses some other address as their username, they could be restricted from updates to their VFS paths as described below. In either case, this possibility should likely be disallowed.

Of note, the function `resolve_home_path()` in `contracts/os/andromeda-vfs/src/state.rs` appears to handle this possibility correctly to prevent spoofing, by first matching on a valid address before doing a username lookup:

```
let user_address = match api.addr_validate(username_or_address) {
        Ok(addr) => addr,
        Err(_e) => USERS.load(storage, username_or_address)?,
    };
```

Recommendation: It is not clear if this has an immediate impact as no explicit attack vector has been found at this time. However, given the complexity of checking all possible places where a username may spoof a valid address, we recommend enforcing that if an end-user wishes to register a username that is a valid address, then it must be their own address.

AND-43
The `amp_receive()` in Kernel Allows Any Smart Contract to Call It

• **Undetermined** ⓘ Fixed

File(s) affected: `contracts/os/andromeda-kernel/src/execute.rs` , `packages/std/src/common/context.rs`

Description:

The `packet.ctx.get_origin() == info.sender` in `amp_receive()` is supposed to stop smart contracts from sending `AMPPkt` to be relayed by the kernel but it appears that the packet is provided by the sender so `info.sender` can set any `packet.ctx.get_origin()` at will and bypass the check using this condition.

This also happens in `packages/std/src/common/context.rs::contains_sender()`

Althogh this cannot be used to spoof any user, this could be a potential way to introduce unexpected attack vectors to the system.

Recommendation: It is recommened to modify the design to ensure only an end-user account can get through this `packet.ctx.get_origin() == info.sender` check.

AND-44
Any sender can get their `AMPMsg` relayed by `kernel`

• **Undetermined** ⓘ Acknowledged

File(s) affected: `contracts/os/andromeda-kernel/src/contract.rs`

Description: Based on the current design, any sender (even if it's an untrusted external code) can get their `AMPMsg` relayed by Kernel. The current control flow:

1. Call `ExecuteMsg::Send` in `kernel` .
2. `kernel` will `handle()` it and send it out to the recepient without check.
3. When an ADO receives this `AMPMsg` , it'll handle it since `kernel` is the `info.sender` .

Recommendation: Add a check to `handle()` to ensure that only ADODB whitelisted ADOs or user accounts can utilize this service.

AND-45
The Possibility of Nesting App and App Components in VFS Path

• **Undetermined** ⓘ Unresolved

File(s) affected: `packages/std/src/os/vfs.rs`

Description: There is a possibility of a nested app and/or app components, such as `/home/user1/app1/app_component1/app2/app_component2` and `/home/user1/app1/app_component1/app_component2` is a risk. It would hurt an app if somehow an attacker introduced a symlink component that resolves to another symlink component in a third-party app. At the very least, there is a possibility of generating misleading information for users.

Recommendation: Add a check at a `vfs` ADO level to ensure that all the VFS paths are as the VFS path examples given in the document (e.g., not nested) and exclude all the unintended edge cases.

AND-46 The `match` Case in `get_asset_string()` Is Incorrect

• **Undetermined** ⓘ Fixed

> **Update**
> This issue is fixed at `01332e0` .

File(s) affected: `packages/std/src/os/adodb.rs`

Description: (This issue is found in commit hash: `e1a8000` during the fix-check stage of the audit.)

The `match` Case in `get_asset_string()` Is Incorrect. It will always return `Some(asset) => Ok(asset)` because `split(':')` does not check if there is a ':' inside the string.

Recommendation: Modify the code to check the existence of ':'.

Definitions

- **High severity** – High-severity issues usually put a large number of users' sensitive information at risk, or are reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
- **Medium severity** – Medium-severity issues tend to put a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or are reasonably likely to lead to moderate financial impact.
- **Low severity** – The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low impact in view of the client's business circumstances.
- **Informational** – The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
- **Undetermined** – The impact of the issue is uncertain.
- **Fixed** – Adjusted program implementation, requirements or constraints to eliminate the risk.
- **Mitigated** – Implemented actions to minimize the impact or likelihood of the risk.
- **Acknowledged** – The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).

Code Documentation

1. In `packages/std/src/ado_base/permissioning.rs` , the function `is_permissioned()` takes a parameter `strict` that is not documented. It should be explained why its usage is not needed in the case of `Blacklisted` . Further, it should be clarified why a `Limited && !strict` permission returns `true` in all cases (effectively making it the same as a `Whitelisted` permission).

Adherence to Best Practices

1. In `contracts/os/andromeda-adodb/src/contract.rs` , `query_ado_versions()` will return versions in ASCII order, which may not be the same as semver version (e.g., `11.0.0` will appear before `2.0.0`).

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Contracts

- `019...920 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/error.rs`
- `4a3...574 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/lib.rs`
- `4bd...fd9 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/os/adodb.rs`
- `4c0...cf1 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/os/aos_querier.rs`
- `8f3...d17 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/os/economics.rs`
- `c92...ef4 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/os/kernel.rs`
- `a95...1ba ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/os/mod.rs`
- `321...ba1 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/os/vfs.rs`

- 78b...697 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/common/context.rs
- e58...76b ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/common/expiration.rs
- ac3...917 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/common/mod.rs
- 7b3...5ef ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/common/queries.rs
- 3ea...05d ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/common/rates.rs
- 967...ad7 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/common/response.rs
- 331...811 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/common/withdraw.rs
- 6ea...b3a ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/amp/addresses.rs
- 513...c7c ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/amp/messages.rs
- 211...8f9 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/amp/mod.rs
- 9b2...fd7 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/amp/recipient.rs
- c10...261 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_contract/app.rs
- 375...84a ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_contract/execute.rs
- 12f...ea6 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_contract/instantiate.rs
- 5f4...647 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_contract/mod.rs
- a01...9f4 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_contract/ownership.rs
- 475...382 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_contract/permissioning.rs
- 8b4...6ca ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_contract/query.rs
- 020...ed0 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_contract/state.rs
- 62f...609 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_contract/withdraw.rs
- 57a...533 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_contract/modules/execute.rs
- 186...e43 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_contract/modules/mod.rs
- 412...73c ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_contract/modules/query.rs
- 0eb...d46 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_base/ado_type.rs
- 175...213 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_base/block_height.rs
- 9eb...0b3 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_base/hooks.rs
- 0a4...4af ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_base/kernel_address.rs
- c63...32c ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_base/mod.rs
- 0dc...4e2 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_base/modules.rs
- 65e...200 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_base/operators.rs

- ed1...f3e ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_base/ownership.rs
- 574...8c3 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_base/permissioning.rs
- a33...902 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_base/version.rs
- 331...811 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/src/ado_base/withdraw.rs
- 04c...201 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/packages/std/macros/src/lib.rs
- e1a...80c ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-vfs/src/contract.rs
- a8f...d86 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-vfs/src/lib.rs
- df6...dd6 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-vfs/src/state.rs
- f8e...306 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-kernel/src/ack.rs
- 105...377 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-kernel/src/contract.rs
- 8ff...eb1 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-kernel/src/execute.rs
- 1c9...2c9 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-kernel/src/ibc.rs
- 837...1e4 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-kernel/src/lib.rs
- 0cf...a80 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-kernel/src/proto.rs
- 0ee...62b ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-kernel/src/query.rs
- 396...476 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-kernel/src/reply.rs
- 56b...484 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-kernel/src/state.rs
- 7b7...5f7 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-kernel/src/sudo.rs
- f47...1f8 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-economics/src/contract.rs
- 1ae...ec5 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-economics/src/lib.rs
- 6e1...cbb ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-economics/src/state.rs
- 60e...649 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-adodb/src/contract.rs
- 328...c2c ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-adodb/src/lib.rs
- eb8...f68 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/os/andromeda-adodb/src/state.rs
- e15...a9e ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/app/andromeda-app-contract/src/contract.rs
- 390...f92 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/app/andromeda-app-contract/src/lib.rs
- 20e...3d6 ./andromeda-core-e1a8000c3e4dfc8921e2c0492cf216177ea5400b/contracts/app/andromeda-app-contract/src/state.rs

Automated Analysis

N/A

Test Suite Results

All tests have passed.

```
running 7 tests
test testing::tests::test_add_address ... ok
test testing::tests::test_remove_address ... ok
test testing::tests::test_execute_hook_blacklist ... ok
test testing::tests::test_andr_get_query ... ok
test testing::tests::test_instantiate ... ok
test testing::tests::test_add_addresses ... ok
test testing::tests::test_execute_hook_whitelist ... ok

test result: ok. 7 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_adodb-4727cf61d52da095)

running 8 tests
test tests::proper_initialization ... ok
test tests::test_remove_action_fees ... ok
test tests::test_get_code_id ... ok
test tests::test_update_publisher ... ok
test tests::test_update_action_fees ... ok
test tests::test_publish ... ok
test tests::test_all_ado_types ... ok
test tests::test_unpublish ... ok

test result: ok. 8 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_app-3675d0111645fda9)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_app_contract-9c18add2aa2c7d18)

running 19 tests
test state::test::test_create_cross_chain_message ... ok
test testing::test_add_app_component ... ok
test testing::test_add_app_component_unauthorized ... ok
test testing::test_claim_ownership_all ... ok
test testing::test_instantiation ... ok
test testing::test_claim_ownership ... ok
test testing::test_claim_ownership_not_found ... ok
test testing::test_instantiation_duplicate_components ... ok
test testing::test_claim_ownership_empty ... ok
test testing::test_add_app_component_duplicate_name ... ok
test testing::test_proxy_message ... ok
test testing::test_claim_ownership_unauth ... ok
test testing::test_proxy_message_unauth ... ok
test testing::test_proxy_message_not_found ... ok
test testing::test_reply_assign_app ... ok
test testing::test_update_address ... ok
test testing::test_empty_instantiation ... ok
test testing::test_update_address_unauth ... ok
test testing::test_update_address_not_found ... ok

test result: ok. 19 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_auction-f94388bf38b85630)

running 39 tests
test state::tests::read_bids_start_after ... ok
test state::tests::read_bids_start_after_limit_too_high ... ok
test state::tests::read_bids_no_params_desc ... ok
```

```
test state::tests::read_bids_no_params ... ok
test state::tests::read_bids_start_after_and_limit_too_high ... ok
test state::tests::read_bids_start_after_limit ... ok
test state::tests::read_bids_limit ... ok
test state::tests::read_bids_start_after_too_high ... ok
test testing::tests::execute_cancel_no_bids ... ok
test testing::tests::execute_cancel_not_token_owner ... ok
test testing::tests::execute_claim_auction_already_claimed ... ok
test testing::tests::execute_bid_below_min_price ... ok
test testing::tests::execute_cancel_auction_ended ... ok
test testing::tests::execute_claim_auction_not_ended ... ok
test testing::tests::execute_claim ... ok
test testing::tests::execute_place_bid_auction_ended ... ok
test testing::tests::execute_place_bid_auction_cancelled ... ok
test testing::tests::execute_cancel_with_bids ... ok
test testing::tests::execute_claim_no_bids ... ok
test testing::tests::execute_place_bid_highest_bidder_cannot_outbid ... ok
test testing::tests::execute_start_auction_zero_duration ... ok
test testing::tests::execute_start_auction_zero_start_time ... ok
test testing::tests::execute_start_auction_start_time_in_past ... ok
test testing::tests::execute_start_auction_after_previous_finished ... ok
test testing::tests::execute_place_bid_auction_not_started ... ok
test testing::tests::execute_update_auction ... ok
test testing::tests::execute_place_bid_bid_smaller_than_highest_bid ... ok
test testing::tests::execute_place_bid_whitelist ... ok
test testing::tests::execute_place_bid_invalid_coins_sent ... ok
test testing::tests::execute_place_bid_token_owner_cannot_bid ... ok
test testing::tests::execute_update_auction_auction_started ... ok
test testing::tests::execute_update_auction_unauthorized ... ok
test testing::tests::execute_update_auction_start_time_in_past ... ok
test testing::tests::execute_update_auction_zero_duration ... ok
test testing::tests::test_auction_instantiate ... ok
test testing::tests::execute_update_auction_zero_start ... ok
test testing::tests::test_execute_place_bid_non_existing_auction ... ok
test testing::tests::execute_place_bid_multiple_bids ... ok
test testing::tests::test_execute_start_auction ... ok

test result: ok. 39 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.01s

     Running unittests src/lib.rs (target/debug/deps/andromeda_cross_chain_swap-64b41f46be3cbb8a)

running 4 tests
test testing::tests::test_instantiate ... ok
test testing::tests::test_swap_and_forward_current_state_failure ... ok
test testing::tests::test_swap_and_forward_invalid_dex ... ok
test testing::tests::test_swap_and_forward_osmo ... ok

test result: ok. 4 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_crowdfund-4a029879b2eb54c0)

running 32 tests
test testing::tests::test_instantiate ... ok
test testing::tests::test_end_sale_not_expired ... ok
test testing::tests::test_addresslist ... ok
test testing::tests::test_end_sale_single_purchase ... ok
test testing::tests::test_end_sale_limit_zero ... ok
test testing::tests::test_end_sale_all_tokens_sold ... ok
test testing::tests::test_mint_sale_conducted_can_mint_after_sale ... ok
test testing::tests::test_mint_owner_not_crowdfund ... ok
test testing::tests::test_mint_sale_conducted_cant_mint_after_sale ... ok
test testing::tests::test_mint_multiple_successful ... ok
test testing::tests::test_mint_unauthorized ... ok
test testing::tests::test_mint_successful ... ok
test testing::tests::test_mint_multiple_exceeds_limit ... ok
test testing::tests::test_mint_sale_started ... ok
test testing::tests::test_purchase_no_funds ... ok
test testing::tests::test_purchase_by_token_id_not_available ... ok
```

```
test testing::tests::test_purchase_not_enough_for_price ... ok
test testing::tests::test_purchase_by_token_id ... ok
test testing::tests::test_purchase_sale_not_started ... ok
test testing::tests::test_integration_conditions_not_met ... ok
test testing::tests::test_purchase_sale_not_ended ... ok
test testing::tests::test_start_sale_expiration_in_past ... ok
test testing::tests::test_purchase_more_than_allowed_per_wallet ... ok
test testing::tests::test_start_sale_max_modified ... ok
test testing::tests::test_purchase_wrong_denom ... ok
test testing::tests::test_start_sale_max_default ... ok
test testing::tests::test_start_sale_no_expiration ... ok
test testing::tests::test_integration_conditions_met ... ok
test testing::tests::test_purchase_not_enough_for_tax ... ok
test testing::tests::test_start_sale_unauthorized ... ok
test testing::tests::test_validate_andr_addresses_regular_address ... ok
test testing::tests::test_multiple_purchases ... ok

test result: ok. 32 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.01s

     Running unittests src/lib.rs (target/debug/deps/andromeda_cw20-db349f4709279cdd)

running 3 tests
test testing::tests::test_andr_query ... ok
test testing::tests::test_send ... ok
test testing::tests::test_transfer ... ok

test result: ok. 3 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_cw20_exchange-52727a7e90b28aff)

running 26 tests
test testing::tests::test_andr_query ... ok
test testing::tests::test_cancel_sale_unauthorised ... ok
test testing::tests::test_cancel_sale_no_sale ... ok
test testing::tests::test_instantiate ... ok
test testing::tests::test_purchase_no_tokens_left_native ... ok
test testing::tests::test_purchase_no_sale ... ok
test testing::tests::test_purchase ... ok
test testing::tests::test_cancel_sale ... ok
test testing::tests::test_purchase_native ... ok
test testing::tests::test_purchase_no_tokens_left ... ok
test testing::tests::test_purchase_not_enough_tokens_native ... ok
test testing::tests::test_query_sale_assets ... ok
test testing::tests::test_start_sale ... ok
test testing::tests::test_start_sale_invalid_token ... ok
test testing::tests::test_purchase_not_enough_sent ... ok
test testing::tests::test_query_token_address ... ok
test testing::tests::test_start_sale_ongoing ... ok
test testing::tests::test_purchase_not_enough_sent_native ... ok
test testing::tests::test_query_sale ... ok
test testing::tests::test_purchase_not_enough_tokens ... ok
test testing::tests::test_start_sale_unauthorised ... ok
test testing::tests::test_purchase_no_sale_native ... ok
test testing::tests::test_start_sale_zero_amount ... ok
test testing::tests::test_purchase_refund ... ok
test testing::tests::test_purchase_native_invalid_coins ... ok
test testing::tests::test_start_sale_zero_exchange_rate ... ok

test result: ok. 26 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_cw20_staking-3846f604c40573d6)

running 23 tests
test testing::tests::test_add_reward_token_duplicate ... ok
test testing::tests::test_add_reward_token_staking_token ... ok
test testing::tests::test_add_reward_token_unauthorized ... ok
test testing::tests::test_add_reward_token ... ok
test testing::tests::test_instantiate_cycle_duration_zero ... ok
```

```
test testing::tests::test_add_reward_token_exceeds_max ... ok
test testing::tests::test_instantiate ... ok
test testing::tests::test_instantiate_exceed_max ... ok
test testing::tests::test_instantiate_invalid_reward_increase ... ok
test testing::tests::test_instantiate_staking_token_as_addtional_reward ... ok
test testing::tests::test_instantiate_end_time_in_past ... ok
test testing::tests::test_instantiate_start_time_in_past ... ok
test testing::tests::test_claim_rewards_allocated ... ok
test testing::tests::test_stake_invalid_token ... ok
test testing::tests::test_claim_rewards ... ok
test testing::tests::test_update_global_indexes_invalid_asset ... ok
test testing::tests::test_update_global_indexes_selective ... ok
test testing::tests::test_receive_cw20_zero_amount ... ok
test testing::tests::test_update_global_indexes ... ok
test testing::tests::test_update_global_indexes_cw20_deposit ... ok
test testing::tests::test_stake_rewards_update ... ok
test testing::tests::test_stake_unstake_tokens ... ok
test testing::tests::test_unstake_rewards_update ... ok

test result: ok. 23 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.01s

     Running unittests src/lib.rs (target/debug/deps/andromeda_cw721-e530b38edee94d1b)

running 10 tests
test testing::test_archived_check ... ok
test testing::test_burn ... ok
test testing::test_agreed_transfer_nft ... ok
test testing::test_archive ... ok
test testing::test_agreed_transfer_nft_wildcard ... ok
test testing::test_batch_mint ... ok
test testing::test_modules ... ok
test testing::test_update_app_contract_invalid_minter ... ok
test testing::test_transfer_agreement ... ok
test testing::test_transfer_nft ... ok

test result: ok. 10 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_data_storage-712f13fec331d211)

running 8 tests
test primitive::tests::try_get_binary ... ok
test primitive::tests::test_parse_error ... ok
test primitive::tests::try_get_bool ... ok
test primitive::tests::try_get_decimal ... ok
test primitive::tests::try_get_object ... ok
test primitive::tests::try_get_string ... ok
test primitive::tests::try_get_vec ... ok
test primitive::tests::try_get_uint128 ... ok

test result: ok. 8 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_economics-026812966296d81a)

running 7 tests
test tests::proper_initialization ... ok
test tests::test_spend_balance ... ok
test tests::test_deposit ... ok
test tests::test_withdraw ... ok
test tests::test_cw20_deposit ... ok
test tests::test_withdraw_cw20 ... ok
test tests::test_pay_fee ... ok

test result: ok. 7 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_ecosystem-135e59748d464a99)

running 0 tests
```

```
test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_finance-329362a85eca0bcc)

running 5 tests
test timelock::tests::test_min_funds_deposited ... ok
test splitter::tests::test_validate_recipient_list ... ok
test timelock::tests::test_add_funds ... ok
test timelock::tests::test_validate_funds_condition ... ok
test timelock::tests::test_validate ... ok

test result: ok. 5 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_fungible_tokens-34135c1f45d3ff2e)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_kernel-8c680f4ce6abfb9c)

running 9 tests
test testing::tests::proper_initialization ... ok
test ibc::tests::test_generate_ibc_denom ... ok
test testing::tests::test_assign_channels_unauthorized ... ok
test testing::tests::test_send_cross_chain_no_channel ... ok
test testing::tests::test_register_user_cross_chain ... ok
test testing::tests::test_assign_channels ... ok
test testing::tests::test_handle_ibc_direct ... ok
test testing::tests::test_create_ado ... ok
test testing::test_handler::test_handle_local ... ok

test result: ok. 9 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_lockdrop-b4b1b3c50946ec6e)

running 27 tests
test testing::tests::test_instantiate ... ok
test testing::tests::test_deposit_native_wrong_denom ... ok
test testing::tests::test_claim_rewards_not_available ... ok
test testing::tests::test_increase_incentives_zero_amount ... ok
test testing::tests::test_deposit_native ... ok
test testing::tests::test_increase_incentives_after_phase_ends ... ok
test testing::tests::test_enable_claims_phase_not_ended ... ok
test testing::tests::test_deposit_native_multiple_denoms ... ok
test testing::tests::test_deposit_native_zero_amount ... ok
test testing::tests::test_enable_claims_no_bootstrap_specified ... ok
test testing::tests::test_instantiate_init_deposit_window_less_than_withdrawal_window ... ok
test testing::tests::test_deposit_native_deposit_window_closed ... ok
test testing::tests::test_claim_rewards ... ok
test testing::tests::test_increase_incentives ... ok
test testing::tests::test_instantiate_init_withdrawal_window_zero ... ok
test testing::tests::test_instantiate_init_timestamp_past ... ok
test testing::tests::test_increase_incentives_invalid_token ... ok
test testing::tests::test_query_withdrawable_percent ... ok
test testing::tests::test_instantiate_init_deposit_window_zero ... ok
test testing::tests::test_withdraw_proceeds_phase_not_ended ... ok
test testing::tests::test_withdraw_proceeds_phase_not_started ... ok
test testing::tests::test_withdraw_native_withdraw_phase_first_half ... ok
test testing::tests::test_withdraw_native_withdrawal_closed ... ok
test testing::tests::test_withdraw_native ... ok
test testing::tests::test_withdraw_native_withdraw_phase_second_half ... ok
test testing::tests::test_withdraw_proceeds_unauthorized ... ok
test testing::tests::test_withdraw_proceeds ... ok

test result: ok. 27 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.01s

     Running unittests src/lib.rs (target/debug/deps/andromeda_macros-5c0c2efda2e870a7)
```

```
running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_marketplace-780c173e8f285907)

running 12 tests
test testing::tests::execute_buy_token_owner_cannot_buy ... ok
test testing::tests::execute_buy_sale_not_open_cancelled ... ok
test testing::tests::execute_start_sale_invalid_price ... ok
test testing::tests::execute_buy_invalid_coins_sent ... ok
test testing::tests::execute_buy_sale_not_open_already_bought ... ok
test testing::tests::execute_buy_with_tax_and_royalty_insufficient_funds ... ok
test testing::tests::execute_buy_with_tax_and_royalty_works ... ok
test testing::tests::execute_buy_works ... ok
test testing::tests::execute_update_sale_invalid_price ... ok
test testing::tests::execute_update_sale_unauthorized ... ok
test testing::tests::test_execute_buy_non_existing_sale ... ok
test testing::tests::test_sale_instantiate ... ok

test result: ok. 12 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_merkle_airdrop-9056580e5d9b999d)

running 9 tests
test testing::tests::cant_burn ... ok
test testing::tests::proper_instantiation ... ok
test testing::tests::can_burn_native ... ok
test testing::tests::claim ... ok
test testing::tests::can_burn ... ok
test testing::tests::claim_native ... ok
test testing::tests::register_merkle_root ... ok
test testing::tests::multiple_claim ... ok
test testing::tests::stage_expires ... ok

test result: ok. 9 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_modules-bd7036936f8ab2c4)

running 1 test
test rates::tests::test_calculate_fee ... ok

test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_non_fungible_tokens-
89777f84d925e2cf)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_primitive-bb86f325b0003df3)

running 10 tests
test testing::tests::test_delete_value ... ok
test testing::tests::test_query_all_key ... ok
test testing::tests::test_instantiation ... ok
test testing::tests::test_set_and_update_value_with_key ... ok
test testing::tests::test_restriction_private ... ok
test testing::tests::test_query_owner_keys ... ok
test testing::tests::test_restriction_public ... ok
test testing::tests::test_restriction_restricted ... ok
test testing::tests::test_set_and_update_value_without_key ... ok
test testing::tests::test_set_object ... ok

test result: ok. 10 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

```
     Running unittests src/lib.rs (target/debug/deps/andromeda_rate_limiting_withdrawals-
81c88805aa481bb7)

running 10 tests
test testing::tests::test_instantiate_works ... ok
test testing::tests::test_deposit_new_account_works ... ok
test testing::tests::test_deposit_existing_account_works ... ok
test testing::tests::test_deposit_zero_funds ... ok
test testing::tests::test_deposit_invalid_funds ... ok
test testing::tests::test_withdraw_account_not_found ... ok
test testing::tests::test_withdraw_funds_locked ... ok
test testing::tests::test_withdraw_over_account_limit ... ok
test testing::tests::test_withdraw_works ... ok
test testing::tests::test_withdraw_over_allowed_limit ... ok

test result: ok. 10 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_rates-8c2a66ed6b96b0ef)

running 4 tests
test testing::tests::test_instantiate_query ... ok
test testing::tests::test_andr_receive ... ok
test testing::tests::test_query_deducted_funds_cw20 ... ok
test testing::tests::test_query_deducted_funds_native ... ok

test result: ok. 4 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_splitter-b561c427376a544b)

running 10 tests
test testing::tests::test_execute_send_error ... ok
test testing::tests::test_execute_update_lock ... ok
test testing::tests::test_execute_send ... ok
test testing::tests::test_execute_send_ado_recipient ... ok
test testing::tests::test_execute_update_recipients ... ok
test testing::tests::test_handle_packet_exit_with_error_true ... ok
test testing::tests::test_query_splitter ... ok
test testing::tests::test_instantiate ... ok
test testing::tests::test_update_app_contract_invalid_recipient ... ok
test testing::tests::test_update_app_contract ... ok

test result: ok. 10 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_std-07c40cb865f92b62)

running 77 tests
test ado_base::withdraw::tests::test_get_amount_amount ... ok
test ado_base::withdraw::tests::test_get_amount_invalid_percentage ... ok
test ado_base::modules::tests::test_validate_uniqueness ... ok
test ado_base::withdraw::tests::test_get_amount_no_withdrawal_type ... ok
test ado_base::withdraw::tests::test_get_too_high_amount ... ok
test ado_base::withdraw::tests::test_get_amount_percentage ... ok
test ado_contract::execute::tests::test_update_app_contract_invalid_module ... ok
test ado_contract::modules::query::tests::test_query_module ... ok
test ado_contract::execute::tests::test_update_app_contract ... ok
test ado_contract::execute::tests::test_register_module_invalid_identifier ... ok
test ado_contract::execute::tests::test_alter_module_invalid_identifier ... ok
test ado_contract::modules::tests::test_execute_alter_module_addr ... ok
test ado_contract::modules::tests::test_execute_alter_module_nonexisting_module ... ok
test ado_contract::modules::tests::test_execute_alter_module_unauthorized ... ok
test ado_contract::modules::tests::test_execute_deregister_module_nonexisting_module ... ok
test ado_contract::modules::tests::test_execute_alter_module_immutable ... ok
test ado_contract::modules::tests::test_execute_deregister_module_unauthorized ... ok
test ado_contract::modules::tests::test_execute_register_module_unauthorized ... ok
test ado_contract::modules::tests::test_execute_register_module_addr ... ok
test ado_contract::modules::tests::test_process_module_response ... ok
test ado_contract::modules::tests::test_load_module_addresses ... ok
test ado_contract::modules::tests::test_execute_deregister_module ... ok
```

```
test ado_contract::permissioning::tests::test_context_permissions_strict ... ok
test ado_contract::permissioning::tests::test_owner_escape_clause ... ok
test ado_contract::permissioning::tests::test_permission_action_unauthorized ... ok
test ado_contract::modules::tests::test_execute_deregister_module_immutable ... ok
test ado_contract::permissioning::tests::test_disable_permissioning_unauthorized ... ok
test ado_contract::permissioning::tests::test_query_permissioned_actions ... ok
test ado_contract::permissioning::tests::test_query_permissions ... ok
test ado_contract::permissioning::tests::test_permissioned_action ... ok
test ado_contract::permissioning::tests::test_remove_permission_unauthorized ... ok
test ado_contract::permissioning::tests::test_strict_permissioning ... ok
test ado_contract::permissioning::tests::test_permission_expiration ... ok
test ado_contract::permissioning::tests::test_set_permission_unauthorized ... ok
test ado_contract::permissioning::tests::test_unpermissioned_action_blacklisted ... ok
test ado_contract::withdraw::tests::test_execute_withdraw_no_funds ... ok
test ado_contract::withdraw::tests::test_execute_withdraw_not_authorized ... ok
test amp::addresses::tests::test_get_chain ... ok
test amp::addresses::tests::test_get_protocol ... ok
test amp::addresses::tests::test_get_raw_path ... ok
test amp::addresses::tests::test_get_root_dir ... ok
test amp::addresses::tests::test_is_local_path ... ok
test amp::addresses::tests::test_is_addr ... ok
test ado_contract::withdraw::tests::test_execute_withdraw_selective ... ok
test ado_contract::withdraw::tests::test_execute_withdraw_cw20 ... ok
test ado_contract::withdraw::tests::test_execute_withdraw_native ... ok
test ado_contract::permissioning::tests::test_context_permissions ... ok
test amp::addresses::tests::test_validate ... ok
test amp::addresses::tests::test_is_vfs ... ok
test amp::messages::tests::test_get_messages_for_recipient ... ok
test amp::messages::tests::test_get_unique_recipients ... ok
test amp::recipient::test::test_generate_amp_msg ... ok
test amp::messages::tests::test_to_json ... ok
test amp::messages::tests::test_to_ibc_hooks_memo ... ok
test amp::recipient::test::test_generate_direct_msg ... ok
test common::test::test_merge_coins ... ok
test amp::messages::tests::test_generate_amp_pkt ... ok
test common::expiration::tests::test_expiration_from_milliseconds ... ok
test common::test::test_parse_struct ... ok
test common::test::test_merge_sub_messages ... ok
test amp::recipient::test::test_generate_msg_cw20 ... ok
test common::test::test_has_coins_merged ... ok
test common::test::test_deduct_funds ... ok
test amp::messages::tests::test_to_sub_msg ... ok
test common::withdraw::tests::test_get_amount_amount ... ok
test amp::messages::tests::test_verify_origin ... ok
test common::withdraw::tests::test_get_amount_no_withdrawal_type ... ok
test common::withdraw::tests::test_get_amount_percentage ... ok
test common::withdraw::tests::test_get_too_high_amount ... ok
test os::adodb::tests::test_get_type ... ok
test os::adodb::tests::test_action_fee_asset ... ok
test os::vfs::test::test_convert_component_name ... ok
test common::withdraw::tests::test_get_amount_invalid_percentage ... ok
test os::adodb::tests::test_get_version ... ok
test os::adodb::tests::test_validate ... ok
test os::vfs::test::test_validate_component_name ... ok
test os::vfs::test::test_validate_path_name ... ok

test result: ok. 77 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.01s

     Running unittests src/lib.rs (target/debug/deps/andromeda_testing-2560e08ade1ce845)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_timelock-c8ef840881b3d558)

running 13 tests
test state::tests::test_get_key ... ok
```

```
test testing::tests::test_execute_release_funds_locked ... ok
test testing::tests::test_execute_hold_funds_escrow_updated ... ok
test testing::tests::test_execute_hold_funds ... ok
test testing::tests::test_execute_release_funds_min_funds_condition ... ok
test testing::tests::test_execute_release_funds_block_condition ... ok
test testing::tests::test_execute_release_funds_no_condition ... ok
test testing::tests::test_execute_release_funds_time_condition ... ok
test testing::tests::test_execute_release_specific_funds_no_funds_locked ... ok
test testing::tests::test_execute_release_multiple_escrows ... ok
test testing::tests::test_execute_release_specific_funds_min_funds_condition ... ok
test testing::tests::test_execute_release_specific_funds_no_condition ... ok
test testing::tests::test_execute_release_specific_funds_time_condition ... ok

test result: ok. 13 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_vault-b146a9d1c4dda8d7)

running 23 tests
test testing::test_deposit_strategy_insufficient_partial_amount ... ok
test testing::test_deposit_insufficient_funds ... ok
test testing::test_deposit_strategy_empty_funds_non_empty_amount ... ok
test testing::test_deposit ... ok
test testing::test_execute_update_strategy ... ok
test testing::test_deposit_strategy ... ok
test testing::test_execute_update_strategy_not_operator ... ok
test testing::test_instantiate ... ok
test testing::test_deposit_strategy_partial_amount ... ok
test testing::test_query_strategy_address ... ok
test testing::test_query_strategy_balance ... ok
test testing::test_query_local_balance ... ok
test testing::test_withdraw_empty ... ok
test testing::test_query_strategy_address_invalid ... ok
test testing::test_withdraw_invalid_strategy ... ok
test testing::test_withdraw_single_no_strategy_amount ... ok
test testing::test_withdraw_multi_no_strategy_insufficientfunds ... ok
test testing::test_withdraw_multi_no_strategy_recipient ... ok
test testing::test_withdraw_single_strategy ... ok
test testing::test_withdraw_multi_no_strategy_mixed ... ok
test testing::test_withdraw_single_no_strategy_insufficientfunds ... ok
test testing::test_withdraw_invalid_withdrawals ... ok
test testing::test_withdraw_single_no_strategy_percentage ... ok

test result: ok. 23 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/andromeda_vesting-0b661cac1dd4caac)

running 41 tests
test testing::tests::test_claim_all_unauthorized ... ok
test testing::tests::test_claim_batch_all_funds_delegated ... ok
test state::tests::test_get_claimable_batches_with_ids ... ok
test testing::tests::test_claim_batch_no_funds_available ... ok
test testing::tests::test_claim_batch_multiple_claims ... ok
test testing::tests::test_claim_batch_all_releases ... ok
test testing::tests::test_claim_batch_not_nice_numbers_multiple_releases ... ok
test testing::tests::test_claim_batch_middle_of_interval ... ok
test testing::tests::test_claim_all ... ok
test testing::tests::test_claim_batch_not_nice_numbers_single_release ... ok
test testing::tests::test_claim_batch_single_claim ... ok
test testing::tests::test_claim_batch_too_high_of_claim ... ok
test testing::tests::test_claim_batch_some_funds_delegated ... ok
test testing::tests::test_claim_batch_still_locked ... ok
test testing::tests::test_claim_batch_unauthorized ... ok
test testing::tests::test_create_batch_invalid_denom ... ok
test testing::tests::test_create_batch_unauthorized ... ok
test testing::tests::test_delegate ... ok
test testing::tests::test_delegate_unauthorized ... ok
test testing::tests::test_create_batch_release_amount_zero ... ok
test testing::tests::test_create_batch_multi_batch_not_supported ... ok
```

```
test testing::tests::test_create_batch_release_unit_zero ... ok
test testing::tests::test_delegate_no_funds ... ok
test testing::tests::test_delegate_more_than_balance ... ok
test testing::tests::test_create_batch_valid_denom_zero_amount ... ok
test testing::tests::test_create_batch_and_delegate ... ok
test testing::tests::test_instantiate ... ok
test testing::tests::test_create_batch_no_funds ... ok
test testing::tests::test_redelegate ... ok
test testing::tests::test_vote ... ok
test testing::tests::test_redelegate_unauthorized ... ok
test testing::tests::test_undelegate_unauthorized ... ok
test testing::tests::test_undelegate_no_funds ... ok
test testing::tests::test_redelegate_no_funds ... ok
test testing::tests::test_vote_unauthorized ... ok
test testing::tests::test_withdraw_rewards_unauthorized ... ok
test testing::tests::test_redelegate_more_than_max ... ok
test testing::tests::test_undelegate ... ok
test testing::tests::test_withdraw_rewards ... ok
test testing::tests::test_undelegate_more_than_max ... ok
test testing::tests::test_create_batch ... ok

test result: ok. 41 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.01s

     Running unittests src/lib.rs (target/debug/deps/andromeda_vfs-a248b58631f9cf69)

running 27 tests
test state::test::test_split_pathname ... ok
test state::test::test_resolve_pathname ... ok
test state::test::test_resolve_lib_path ... ok
test state::test::test_resolve_path_too_long ... ok
test state::test::test_resolve_home_path ... ok
test state::test::test_resolve_path_loop ... ok
test testing::tests::proper_initialization ... ok
test state::test::test_add_symlink_looping_reference ... ok
test testing::tests::test_get_library ... ok
test testing::tests::test_get_username ... ok
test testing::tests::test_add_child_not_app_contract ... ok
test state::test::test_resolve_symlink ... ok
test testing::tests::test_get_paths ... ok
test testing::tests::test_register_user_cross_chain ... ok
test testing::tests::test_register_user_valid_cosmwasm_address_user ... ok
test testing::tests::test_register_user_valid_cosmwasm_address ... ok
test testing::tests::test_register_user_foreign_chain ... ok
test testing::tests::test_register_user_unauthorized ... ok
test testing::tests::test_override_add_child ... ok
test testing::tests::test_register_user_already_registered ... ok
test testing::tests::test_add_child ... ok
test state::test::test_validate_username ... ok
test testing::tests::test_register_user_duplicate ... ok
test testing::tests::test_register_user ... ok
test testing::tests::test_add_path ... ok
test testing::tests::test_get_subdir ... ok
test testing::tests::test_add_symlink ... ok

test result: ok. 27 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.01s

     Running unittests src/lib.rs (target/debug/deps/andromeda_weighted_distribution_splitter-
42712c1f38d8f45e)

running 24 tests
test testing::tests::test_execute_add_recipient_unauthorized ... ok
test testing::tests::test_execute_add_recipient_invalid_weight ... ok
test testing::tests::test_execute_remove_recipient_contract_locked ... ok
test testing::tests::test_execute_add_recipient_duplicate_recipient ... ok
test testing::tests::test_execute_remove_recipient_not_on_list ... ok
test testing::tests::test_execute_add_recipient_locked_contract ... ok
test testing::tests::test_execute_add_recipient ... ok
test testing::tests::test_execute_remove_recipient ... ok
```

```
test testing::tests::test_execute_update_lock_already_locked ... ok
test testing::tests::test_execute_remove_recipient_unauthorized ... ok
test testing::tests::test_execute_update_lock_too_long ... ok
test testing::tests::test_execute_update_lock ... ok
test testing::tests::test_execute_update_lock_too_short ... ok
test testing::tests::test_execute_update_recipients_contract_locked ... ok
test testing::tests::test_execute_update_lock_unauthorized ... ok
test testing::tests::test_execute_update_recipients ... ok
test testing::tests::test_execute_update_recipients_invalid_weight ... ok
test testing::tests::test_execute_update_recipients_unauthorized ... ok
test testing::tests::test_instantiate ... ok
test testing::tests::test_update_recipient_weight ... ok
test testing::tests::test_update_recipient_weight_locked_contract ... ok
test testing::tests::test_update_recipient_weight_invalid_weight ... ok
test testing::tests::test_update_app_contract ... ok
test testing::tests::test_update_recipient_weight_user_not_found ... ok

test result: ok. 24 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running unittests src/lib.rs (target/debug/deps/tests_integration-1ea4a62a5d22c1f9)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

     Running tests/auction_app.rs (target/debug/deps/auction_app-fe27da9fd2d49421)

running 1 test
test test_auction_app ... ok

test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.01s

     Running tests/crowdfund_app.rs (target/debug/deps/crowdfund_app-dd86ea627e22fe82)

running 1 test
test test_crowdfund_app ... ok

test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.02s

     Running tests/kernel.rs (target/debug/deps/kernel-35c518029e0010ec)

running 1 test
test kernel ... ok

test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.01s

     Running tests/marketplace_app.rs (target/debug/deps/marketplace_app-dad5cb6ff621985a)

running 1 test
test test_marketplace_app ... ok

test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.01s

     Running tests/mod.rs (target/debug/deps/mod-c932d03e644f925d)

running 4 tests
test kernel::kernel ... ok
test auction_app::test_auction_app ... ok
test marketplace_app::test_marketplace_app ... ok
test crowdfund_app::test_crowdfund_app ... ok

test result: ok. 4 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.02s

     Running tests/primtive.rs (target/debug/deps/primtive-102de0a62fc4d3d8)

running 1 test
test test_primtive ... ok
```

```
test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-address-list

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-adodb

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-app

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-app-contract

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-auction

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-cross-chain-swap

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-crowdfund

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-cw20

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-cw20-exchange

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-cw20-staking

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-cw721

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

```
   Doc-tests andromeda-data-storage

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-economics

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-ecosystem

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-finance

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-fungible-tokens

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-kernel

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-lockdrop

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-macros

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-marketplace

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-merkle-airdrop

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-modules

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-non-fungible-tokens
```

```
running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-primitive

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-rate-limiting-withdrawals

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-rates

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-splitter

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-std

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-testing

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-timelock

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-vault

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-vesting

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-vfs

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests andromeda-weighted-distribution-splitter

running 0 tests
```

```
test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests tests-integration

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

Code Coverage

The current coverage socre is about 76.25%. We recommend adding more tests to increase the coverage and to avoid functional bugs that are not necessarily security issues.

2024-02-21T06:39:37.553842Z INFO cargo_tarpaulin::report: Coverage Results: || Uncovered Lines: || contracts/app/andromeda-app-contract/src/contract.rs: 41, 43, 69-74, 77-78, 80-82, 87-90, 92, 133, 136-140, 142, 151-152, 157-159, 173-174, 194, 199, 201, 204-205, 207, 209-212, 217-220, 224, 227, 229, 237, 240-242 || contracts/app/andromeda-app-contract/src/execute.rs: 53, 55, 59-60, 62, 65-66, 68-69, 71, 217 || contracts/app/andromeda-app-contract/src/query.rs: 21-22, 25-27, 30-32, 34 || contracts/app/andromeda-app-contract/src/state.rs: 50, 53-55, 57-59, 63, 111-114, 116-117, 119, 166, 168 || contracts/data-storage/andromeda-primitive/src/contract.rs: 59-60, 67, 69, 72-73, 75, 77-80, 85-88, 92, 95, 97, 106 || contracts/data-storage/andromeda-primitive/src/execute.rs: 15, 18, 22, 26-30, 32-33, 35 || contracts/data-storage/andromeda-primitive/src/mock.rs: 33, 35-36 || contracts/ecosystem/andromeda-vault/src/contract.rs: 57-60, 64, 95-96, 344, 346, 385, 387, 390-391, 393, 395-398, 403-406, 410, 413, 415, 427, 446 || contracts/finance/andromeda-cross-chain-swap/src/contract.rs: 59, 61-62, 64, 66-70, 73-78, 80-81, 83, 86-88, 90-91, 93, 95-101, 103, 105, 110-112, 116-117, 119-120, 135-136, 161, 184, 220, 222, 225-226, 228, 230-233, 238-241, 245, 248, 250, 254-255 || contracts/finance/andromeda-cross-chain-swap/src/dex.rs: 16-19, 23-24, 26-27 || contracts/finance/andromeda-rate-limiting-withdrawals/src/contract.rs: 47-52, 116-117, 127, 210-212, 216-219, 223, 228, 232, 280, 282, 285-286, 288, 290-293, 298-301, 305, 308, 310, 314-318, 320, 324-327, 329, 333-335 || contracts/finance/andromeda-splitter/src/contract.rs: 41, 43, 92-93, 129, 131-132, 221, 223, 226, 228, 244, 246, 253, 255, 280, 282, 285-286, 288, 290-293, 298-301, 305, 308, 310 || contracts/finance/andromeda-timelock/src/contract.rs: 27, 33, 35-46, 50, 63-64, 86, 199, 201, 204-205, 207, 209-212, 217-220, 224, 227, 229, 238, 248, 252, 258-261, 263-264 || contracts/finance/andromeda-vesting/src/contract.rs: 78-79, 112, 269, 493, 496, 544, 546, 549-550, 552, 554-557, 562-565, 569, 572, 574, 580, 585, 589-591 || contracts/finance/andromeda-weighted-distribution-splitter/src/contract.rs: 42, 44, 49, 51, 54, 57-58, 99-100, 116, 131, 133, 179, 181, 190, 192, 212, 214, 227-228, 230-233, 237-239, 242-245, 248-250, 255-262, 266, 268-269, 271, 273-276, 307-309, 329, 331, 343, 345, 368, 370, 449, 4 51, 454-455, 457, 459-462, 467-470, 474, 477, 479, 483-487, 491-493, 495, 498-501, 504-506, 508, 511-513, 518-519, 521 || contracts/fungible-tokens/andromeda-cw20/src/contract.rs: 73-74, 95, 101, 103-106, 135, 155, 161, 164-165, 168, 171-172, 175, 178, 180, 183, 187, 216, 240, 246, 249, 253, 267, 269, 274, 277, 284, 286, 289-290, 292, 294-297, 302-305, 309, 312, 314, 322 || contracts/fungible-tokens/andromeda-cw20-exchange/src/contract.rs: 64-67, 71, 84-85, 96, 352, 354, 357-358, 360, 362-365, 370-373, 377, 380, 382, 393 || contracts/fungible-tokens/andromeda-cw20-staking/src/allocated_rewards.rs: 19, 57, 62, 66, 120 || contracts/fungible-tokens/andromeda-cw20-staking/src/contract.rs: 124-125, 173, 219, 388, 457, 557, 627-628, 633-634, 638-639, 642-643, 709-710, 714, 716, 719-720, 722, 724-727, 732-735, 739, 742, 744 || contracts/fungible-tokens/andromeda-lockdrop/src/contract.rs: 106-107, 136, 141, 143, 146-147, 149, 151-154, 159-162, 166, 169, 171, 200 || contracts/fungible-tokens/andromeda-merkle-airdrop/src/contract.rs: 82-83, 102, 115, 117, 177, 187, 233, 235, 250, 252, 296, 349, 351, 354-355, 357, 359-362, 367-370, 374, 377, 379 || contracts/modules/andromeda-address-list/src/contract.rs: 60-61, 121, 123, 144, 146, 149-150, 152, 154-157, 162-165, 169, 172, 174, 182-183, 202-204 || contracts/modules/andromeda-address-list/src/state.rs: 15 || contracts/modules/andromeda-rates/src/contract.rs: 67-68, 88, 90, 100, 102, 105-106, 108, 110-113, 118-121, 125, 128, 130, 138 || contracts/non-fungible-tokens/andromeda-auction/src/contract.rs: 562-563, 608, 618, 622, 626, 630, 634, 639, 641-642, 644, 647, 653, 656, 660, 663, 669, 672-674, 676, 678, 694-695, 699, 705, 735, 763, 765, 768-769, 771, 773-776, 781-784, 788, 791, 793 || contracts/non-fungible-tokens/andromeda-auction/src/state.rs: 81, 87-88, 90, 93-94, 98-100, 102 || contracts/non-fungible-tokens/andromeda-crowdfund/src/contract.rs: 80-81, 98-99, 497, 499, 501, 503, 605, 764, 818-819, 824, 828-829, 832-833, 849, 851, 854-855, 857, 859-862, 867-870, 874, 877, 879 || contracts/non-fungible-tokens/andromeda-cw721/src/contract.rs: 92-94, 105, 113, 117, 119, 132-134, 166, 173, 177-178, 191, 200, 223, 244, 253, 297-300, 306, 308, 393, 401-403, 405, 421, 423, 431, 443, 445, 503, 507, 521-523, 527, 529, 532-533, 535, 537-540, 545-548, 552, 555, 557 || contracts/non-fungible-tokens/andromeda-cw721/src/mock.rs: 77, 84, 88 || contracts/non-fungible-tokens/andromeda-marketplace/src/contract.rs: 200-201, 203-204, 207-213, 258, 262, 324, 326, 330, 332, 430-432, 436-437, 441, 451, 455, 460-464, 467, 470, 476, 479, 484, 486-487, 489, 492-494, 514, 516, 519-520, 522, 524-527, 532-535, 539, 542, 544 || contracts/non-fungible-tokens/andromeda-marketplace/src/state.rs: 53, 55-58, 91, 97-98, 100, 103-104, 108-110, 112 || contracts/os/andromeda-adodb/src/contract.rs: 42-45, 49, 96, 98, 101-102, 104, 106-109, 114-117, 121, 124, 126, 134-135, 137, 141, 146-148, 154-156, 158-159 || contracts/os/andromeda-adodb/src/execute.rs: 56, 58-59, 67, 69, 74, 76-77, 112, 115-116, 119, 121-122, 129, 131 || contracts/os/andromeda-adodb/src/mock.rs: 47 || contracts/os/andromeda-adodb/src/query.rs: 18-20, 23, 26, 30-31, 33, 38-40, 61, 67-69, 72-73, 75-76, 78, 80-83, 85, 102-105, 107-109, 113, 118-119, 122, 127-128 || contracts/os/andromeda-adodb/src/state.rs: 29, 31, 53-54, 66, 68, 79 || contracts/os/andromeda-economics/src/contract.rs: 46, 53-55, 57-58, 60, 101, 103, 106-107, 109, 111-114, 119-122, 126, 129, 131, 135, 137-138 || contracts/os/andromeda-economics/src/execute.rs: 146, 194, 224 || contracts/os/andromeda-economics/src/query.rs: 6-9, 11 || contracts/os/andromeda-kernel/src/ack.rs: 12-14, 17-19 || contracts/os/andromeda-kernel/src/contract.rs: 62, 110, 116-118, 124-125, 131, 133, 136-137, 139, 141-144, 149-152, 156, 159, 161, 164-165, 172-173, 175-176 || contracts/os/andromeda-kernel/src/execute.rs: 28, 43, 45, 47-48, 55-56, 78, 80, 92, 94, 99, 125, 128, 216-217, 295-297, 299-302, 306, 309, 311, 313, 482-483, 487, 492-493, 512-513, 520-521, 543, 552-556, 559-561, 563-564, 567-569, 572-575, 577, 580-581, 585-586, 588-589, 592-599, 60 1-604, 606-607, 609-610, 612 || contracts/os/andromeda-kernel/src/ibc.rs: 48, 57, 61, 66, 70, 75, 77,

79, 81, 85, 90, 92, 94, 98, 106-111, 116, 121, 124, 129-132, 134, 138, 144-147, 149-153, 155, 160-161, 166, 172, 181, 186, 189, 192-195, 197, 199-201, 204, 211-213, 215-219, 231-236, 241-242, 255, 266-267, 273-274 , 276, 279, 281-283, 285, 287-295, 307-308, 311, 313-316, 322, 324-326, 329-332, 336, 339-341, 343-347, 352-353, 357-360, 363-365, 368-370, 373 || contracts/os/andromeda-kernel/src/query.rs: 24, 28, 32-38, 41, 43, 46-48 || contracts/os/andromeda-kernel/src/reply.rs: 56, 60-62, 66, 68, 71-72, 74-77, 81-84, 87-93, 97, 99-101 || contracts/os/andromeda-kernel/src/sudo.rs: 13, 20, 23, 25, 27, 29, 31, 33, 37-38, 40-43, 46, 48-49, 52, 57, 60, 62, 64, 67, 70-71, 73-76, 79-81 || contracts/os/andromeda-vfs/src/contract.rs: 43-46, 50, 80, 91, 93, 96-97, 99, 101-104, 109-112, 116, 119, 121, 124-125 || contracts/os/andromeda-vfs/src/execute.rs: 68, 73, 131-132, 151, 153, 205, 210-215, 218-219, 221, 223-226 || contracts/os/andromeda-vfs/src/state.rs: 85-86, 146, 183, 185 || packages/andromeda-app/src/app.rs: 20-21, 30, 34-36, 38-39, 73, 76 || packages/andromeda-data-storage/src/primitive.rs: 72-73, 78-79, 84-85, 90-91, 96-97, 102-103, 108-109, 114-115, 120-121, 163-166, 170-173, 187 || packages/andromeda-ecosystem/src/vault.rs: 40-41 || packages/andromeda-fungible-tokens/src/cw20.rs: 102-104, 108, 117, 126, 135, 144, 157, 166-167, 231-232, 234, 239, 248, 253 || packages/andromeda-modules/src/rates.rs: 140, 143 || packages/andromeda-non-fungible-tokens/src/cw721.rs: 141-143, 150, 159, 168, 171, 175, 187-188, 274, 304, 307, 309, 318, 325 || packages/andromeda-testing/src/mock_contract.rs: 13-14 || packages/std/macros/src/lib.rs: 7, 11-12, 14-15, 17, 21, 23, 25, 39, 42-43, 63-65, 85-87, 102-103, 107, 110-111, 115, 127-134, 137-139, 144, 149, 154-160, 162, 167, 172, 180, 183-184, 217-219, 232-234, 244 || packages/std/src/ado_base/hooks.rs: 31, 33-35 || packages/std/src/ado_base/permissioning.rs: 53-54, 83-84, 103-107, 117, 120, 126-130, 132, 135-137, 139, 142-144, 146, 150 || packages/std/src/ado_base/withdraw.rs: 41-44 || packages/std/src/ado_contract/app.rs: 29, 35 || packages/std/src/ado_contract/execute.rs: 29, 54-55, 61-64, 68-71, 76-77, 82, 85-86, 88, 110, 119, 121, 138, 174-179, 199-200, 223 || packages/std/src/ado_contract/instantiate.rs: 18-20, 24, 28, 39, 45, 47, 49, 51 || packages/std/src/ado_contract/modules/execute.rs: 7, 12-15, 17, 59-60, 77-78 || packages/std/src/ado_contract/modules/mod.rs: 46, 60, 138, 157-158, 192, 227 || packages/std/src/ado_contract/modules/query.rs: 19, 21 || packages/std/src/ado_contract/ownership.rs: 15, 35 || packages/std/src/ado_contract/permissioning.rs: 47-50, 82, 96, 108, 138, 140, 145, 147-152, 190, 195, 229-230, 232-235, 238-242, 257-258, 261-264, 280-281, 311-314, 330-331, 334, 342 || packages/std/src/ado_contract/query.rs: 31-33, 35-37, 39-40, 42-43, 45, 47, 49-51, 53-59, 61, 63, 79, 84-85, 90-92, 96-99, 101-102, 107-109, 113, 117-118, 122-124, 128-130 || packages/std/src/ado_contract/withdraw.rs: 11, 18-20, 22-23, 25, 29, 35-37, 40-41, 53, 55, 64, 66, 84, 92, 104, 113-114 || packages/std/src/amp/addresses.rs: 32-33, 114, 233-234, 245-246, 269-270 || packages/std/src/amp/messages.rs: 31, 54, 61-62, 70, 72-74, 76, 121, 123-125, 162, 175, 184, 190-194, 332, 337-341, 365-368 || packages/std/src/amp/recipient.rs: 43-44 || packages/std/src/common/context.rs: 32-34 || packages/std/src/common/mod.rs: 38-40, 49, 53-56, 82-83 || packages/std/src/common/queries.rs: 9-12 || packages/std/src/common/rates.rs: 27 || packages/std/src/common/withdraw.rs: 43 || packages/std/src/error.rs: 612-614, 627, 633, 639-641, 645, 654-655, 660-661, 666, 672, 682-683 || packages/std/src/os/adodb.rs: 62, 64-67, 86, 88-89, 106-107, 176-177, 244-245 || packages/std/src/os/aos_querier.rs: 62, 67-69, 82, 102, 107-108, 110-112, 154, 181, 194, 209 || packages/std/src/testing/mock_querier.rs: 99-102, 147-150, 163, 202, 207, 209-211, 213, 215, 232-234, 236-237, 239-240, 242-243, 251, 265-267, 273, 335, 339-343, 345-346, 349-350, 352-353, 364, 368-371, 373-376, 382-383, 400, 425, 428, 448, 457-459, 461-462, 464, 467, 484, 487, 503, 511, 517, 520, 524-527, 529, 535 || Tested/Total Lines: || contracts/app/andromeda-app-contract/src/contract.rs: 71/124 || contracts/app/andromeda-app-contract/src/execute.rs: 96/107 || contracts/app/andromeda-app-contract/src/mock.rs: 8/8 || contracts/app/andromeda-app-contract/src/query.rs: 6/15 || contracts/app/andromeda-app-contract/src/reply.rs: 15/15 || contracts/app/andromeda-app-contract/src/state.rs: 74/91 || contracts/data-storage/andromeda-primitive/src/contract.rs: 24/43 || contracts/data-storage/andromeda-primitive/src/execute.rs: 32/43 || contracts/data-storage/andromeda-primitive/src/mock.rs: 5/8 || contracts/data-storage/andromeda-primitive/src/query.rs: 31/31 || contracts/ecosystem/andromeda-vault/src/contract.rs: 207/234 || contracts/ecosystem/andromeda-vault/src/mock.rs: 6/6 || contracts/finance/andromeda-cross-chain-swap/src/contract.rs: 47/109 || contracts/finance/andromeda-cross-chain-swap/src/dex.rs: 8/16 || contracts/finance/andromeda-rate-limiting-withdrawals/src/contract.rs: 91/140 || contracts/finance/andromeda-splitter/src/contract.rs: 117/148 || contracts/finance/andromeda-splitter/src/mock.rs: 5/5 || contracts/finance/andromeda-timelock/src/contract.rs: 72/115 || contracts/finance/andromeda-timelock/src/state.rs: 16/16 || contracts/finance/andromeda-vesting/src/contract.rs: 292/319 || contracts/finance/andromeda-vesting/src/state.rs: 32/32 || contracts/finance/andromeda-weighted-distribution-splitter/src/contract.rs: 128/226 || contracts/fungible-tokens/andromeda-cw20/src/contract.rs: 93/140 || contracts/fungible-tokens/andromeda-cw20-exchange/src/contract.rs: 165/190 || contracts/fungible-tokens/andromeda-cw20-staking/src/allocated_rewards.rs: 49/54 || contracts/fungible-tokens/andromeda-cw20-staking/src/contract.rs: 322/355 || contracts/fungible-tokens/andromeda-cw20-staking/src/state.rs: 16/16 || contracts/fungible-tokens/andromeda-lockdrop/src/contract.rs: 262/282 || contracts/fungible-tokens/andromeda-merkle-airdrop/src/contract.rs: 150/178 || contracts/modules/andromeda-address-list/src/contract.rs: 77/102 || contracts/modules/andromeda-address-list/src/mock.rs: 6/6 || contracts/modules/andromeda-address-list/src/state.rs: 6/7 || contracts/modules/andromeda-rates/src/contract.rs: 86/107 || contracts/modules/andromeda-rates/src/mock.rs: 4/4 || contracts/non-fungible-tokens/andromeda-auction/src/contract.rs: 323/367 || contracts/non-fungible-tokens/andromeda-auction/src/mock.rs: 12/12 || contracts/non-fungible-tokens/andromeda-auction/src/state.rs: 25/35 || contracts/non-fungible-tokens/andromeda-crowdfund/src/contract.rs: 403/436 || contracts/non-fungible-tokens/andromeda-crowdfund/src/mock.rs: 15/15 || contracts/non-fungible-tokens/andromeda-crowdfund/src/state.rs: 6/6 || contracts/non-fungible-tokens/andromeda-cw721/src/contract.rs: 217/273 || contracts/non-fungible-tokens/andromeda-cw721/src/mock.rs: 13/16 || contracts/non-fungible-tokens/andromeda-cw721/src/state.rs: 3/3 || contracts/non-fungible-tokens/andromeda-marketplace/src/contract.rs: 190/247 || contracts/non-fungible-tokens/andromeda-marketplace/src/mock.rs: 10/10 || contracts/non-fungible-tokens/andromeda-marketplace/src/state.rs: 10/25 || contracts/os/andromeda-adodb/src/contract.rs: 29/62 || contracts/os/andromeda-adodb/src/execute.rs: 109/125 || contracts/os/andromeda-adodb/src/mock.rs: 10/11 || contracts/os/andromeda-adodb/src/query.rs: 9/47 || contracts/os/andromeda-adodb/src/state.rs: 38/45 || contracts/os/andromeda-economics/src/contract.rs: 28/54 || contracts/os/andromeda-economics/src/execute.rs: 137/140 || contracts/os/andromeda-economics/src/mock.rs: 4/4 || contracts/os/andromeda-economics/src/query.rs: 0/5 || contracts/os/andromeda-kernel/src/ack.rs: 0/6 || contracts/os/andromeda-kernel/src/contract.rs: 36/65 || contracts/os/andromeda-kernel/src/execute.rs: 196/274 || contracts/os/andromeda-kernel/src/ibc.rs: 3/131 || contracts/os/andromeda-kernel/src/mock.rs: 9/9 || contracts/os/andromeda-kernel/src/query.rs: 9/23 || contracts/os/andromeda-kernel/src/reply.rs: 11/38 || contracts/os/andromeda-kernel/src/sudo.rs: 0/32 || contracts/os/andromeda-vfs/src/contract.rs: 31/55 || contracts/os/andromeda-vfs/src/execute.rs: 114/134 || contracts/os/andromeda-vfs/src/mock.rs: 6/6 || contracts/os/andromeda-

vfs/src/query.rs: 20/20 || contracts/os/andromeda-vfs/src/state.rs: 116/121 || packages/andromeda-app/src/app.rs: 16/26 || packages/andromeda-data-storage/src/primitive.rs: 30/57 || packages/andromeda-ecosystem/src/vault.rs: 19/21 || packages/andromeda-finance/src/splitter.rs: 15/15 || packages/andromeda-finance/src/timelock.rs: 41/41 || packages/andromeda-fungible-tokens/src/cw20.rs: 7/24 || packages/andromeda-fungible-tokens/src/cw20_staking.rs: 30/30 || packages/andromeda-modules/src/rates.rs: 29/31 || packages/andromeda-non-fungible-tokens/src/auction.rs: 14/14 || packages/andromeda-non-fungible-tokens/src/cw721.rs: 5/21 || packages/andromeda-testing/src/mock.rs: 61/61 || packages/andromeda-testing/src/mock_contract.rs: 10/12 || packages/std/macros/src/lib.rs: 0/57 || packages/std/src/ado_base/hooks.rs: 0/4 || packages/std/src/ado_base/modules.rs: 12/12 || packages/std/src/ado_base/permissioning.rs: 27/53 || packages/std/src/ado_base/withdraw.rs: 10/14 || packages/std/src/ado_contract/app.rs: 10/12 || packages/std/src/ado_contract/execute.rs: 68/98 || packages/std/src/ado_contract/instantiate.rs: 14/24 || packages/std/src/ado_contract/modules/execute.rs: 26/36 || packages/std/src/ado_contract/modules/mod.rs: 95/102 || packages/std/src/ado_contract/modules/query.rs: 10/12 || packages/std/src/ado_contract/ownership.rs: 22/24 || packages/std/src/ado_contract/permissioning.rs: 150/195 || packages/std/src/ado_contract/query.rs: 9/57 || packages/std/src/ado_contract/state.rs: 13/13 || packages/std/src/ado_contract/withdraw.rs: 45/67 || packages/std/src/amp/addresses.rs: 85/94 || packages/std/src/amp/messages.rs: 80/111 || packages/std/src/amp/recipient.rs: 45/47 || packages/std/src/common/context.rs: 7/10 || packages/std/src/common/expiration.rs: 6/6 || packages/std/src/common/mod.rs: 60/70 || packages/std/src/common/queries.rs: 0/4 || packages/std/src/common/rates.rs: 7/8 || packages/std/src/common/response.rs: 3/3 || packages/std/src/common/withdraw.rs: 13/14 || packages/std/src/error.rs: 0/17 || packages/std/src/os/adodb.rs: 40/54 || packages/std/src/os/aos_querier.rs: 58/73 || packages/std/src/os/kernel.rs: 3/3 || packages/std/src/os/vfs.rs: 40/40 || packages/std/src/testing/mock_querier.rs: 129/205 || tests-integration/tests/primtive.rs: 12/12 || 76.25% coverage, 6137/8048 lines covered

Changelog

- 2024-01-30 - Initial Report
- 2024-03-05 - Final Report

About Quantstamp

Quantstamp is a global leader in blockchain security. Founded in 2017, Quantstamp's mission is to securely onboard the next billion users to Web3 through its best-in-class Web3 security products and services.

Quantstamp's team consists of cybersecurity experts hailing from globally recognized organizations including Microsoft, AWS, BMW, Meta, and the Ethereum Foundation. Quantstamp engineers hold PhDs or advanced computer science degrees, with decades of combined experience in formal verification, static analysis, blockchain audits, penetration testing, and original leading-edge research.

To date, Quantstamp has performed more than 500 audits and secured over $200 billion in digital asset risk from hackers. Quantstamp has worked with a diverse range of customers, including startups, category leaders and financial institutions. Brands that Quantstamp has worked with include Ethereum 2.0, Binance, Visa, PayPal, Polygon, Avalanche, Curve, Solana, Compound, Lido, MakerDAO, Arbitrum, OpenSea and the World Economic Forum.

Quantstamp's collaborations and partnerships showcase our commitment to world-class research, development and security. We're honored to work with some of the top names in the industry and proud to secure the future of web3.

Notable Collaborations & Customers:

- Blockchains: Ethereum 2.0, Near, Flow, Avalanche, Solana, Cardano, Binance Smart Chain, Hedera Hashgraph, Tezos
- DeFi: Curve, Compound, Maker, Lido, Polygon, Arbitrum, SushiSwap
- NFT: OpenSea, Parallel, Dapper Labs, Decentraland, Sandbox, Axie Infinity, Illuvium, NBA Top Shot, Zora
- Academic institutions: National University of Singapore, MIT

site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on any website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any output generated by such software.

Disclaimer

The review and this report are provided on an as-is, where-is, and as-available basis. To the fullest extent permitted by law, Quantstamp disclaims all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. You agree that your access and/or use of the report and other results of the review, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE. This report is based on the scope of materials and documentation provided for a limited review at the time provided. You acknowledge that Blockchain technology remains under development and is subject to unknown risks and flaws and, as such, the report may not be complete or inclusive of all vulnerabilities. The review is limited to the materials identified in the report and does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. The report does not indicate the endorsement by Quantstamp of any particular project or team, nor guarantee its security, and may not be represented as such. No third party is entitled to rely on the report in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. Quantstamp does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party, or any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, or any related services and products, any hyperlinked websites, or any other websites or mobile applications, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third party. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.