Redacted Dinero Infrastructure Security Review

Auditors

Parithosh, Lead Security Researcher

Rafael Matias, Lead Security Researcher

Report prepared by: Lucas Goiriz

December 19, 2023

Contents

1 About Spearbit

Spearbit is a decentralized network of expert security engineers offering reviews and other security related services to Web3 projects with the goal of creating a stronger ecosystem. Our network has experience on every part of the blockchain technology stack, including but not limited to protocol design, smart contracts and the Solidity compiler. Spearbit brings in untapped security talent by enabling expert freelance auditors seeking flexibility to work on interesting projects together.

Learn more about us at spearbit.com

2 Introduction

The Redacted ecosystem is a product suite of smart contracts empowering on-chain liquidity, governance, and cash flow for DeFi protocols.

Disclaimer: This security review does not guarantee against a hack. It is a snapshot in time of dinero-infra according to the specific commit. Any modifications to the code will require a new security review.

The following directories are part of the scope:

Directory	Description
`cdk-dinero-keeper`	Keeper functions that watch and interact with the dinero contracts. Infrastructure as code to run those functions on AWS Lambda
`linux-ec2-cdk`	Infrastructure as code to prepare a VM to run a dineronode
`dineronode`	Node software
`dineronode-install`	Node installer. Packaging and configuration scripts

3 Risk classification

Severity level	**Impact: High**	**Impact: Medium**	**Impact: Low**
Likelihood: high	Critical	High	Medium
Likelihood: medium	High	Medium	Low
Likelihood: low	Medium	Low	Low

3.1 Impact

- High - leads to a loss of a significant portion (>10%) of assets in the protocol, or significant harm to a majority of users.
- Medium - global losses <10% or losses to only a subset of users, but still unacceptable.
- Low - losses will be annoying but bearable--applies to things like griefing attacks that can be easily repaired or even gas inefficiencies.

3.2 Likelihood

- High - almost certain to happen, easy to perform, or not easy but highly incentivized
- Medium - only conditionally possible or incentivized, but still relatively likely
- Low - requires stars to align, or little-to-no incentive

3.3 Action required for severity levels

- Critical - Must fix as soon as possible (if already deployed)
- High - Must fix (before deployment if not already deployed)
- Medium - Should fix
- Low - Could fix

4 Executive Summary

Over the course of 10 days in total, Redacted engaged with Spearbit to review the dinero-infra protocol. In this period of time a total of **29** issues were found.

Summary

Project Name	Redacted
Repository	dinero-infra
Commit	33987b...cb1e6b
Type of Project	Infrastructure
Audit Timeline	Nov 8 to Nov 25
Two week fix period	Nov 25 - Dec 13

Issues Found

Severity	Count	Fixed	Acknowledged
Critical Risk	1	1	0
High Risk	0	0	0
Medium Risk	2	2	0
Low Risk	7	5	2
Gas Optimizations	0	0	0
Informational	19	8	11
Total	**29**	**16**	**13**

5 Findings

5.1 Critical Risk

5.1.1 Open ports to the internet

Severity: Critical Risk

Context: linux-ec2-cdk/lib/linux-ec2-cdk-stack.ts#L49-L83

Description: The following ports will be open from the internet (`0.0.0.0/0`) and would allow anyone to access any service running under them.

- 443 tcp - https
- 9100-9104 tcp - beacon Node metrics port
- 9091 tcp - prometheus
- 3100 tcp - grafana
- 8545 tcp - execution layer rpc
- 9001 tcp - prometheus
- 5052 - beacon API

Recommendation: Remove the ports from the security group and keep public facing ports to the minimum (the API ports currently exposed can be easily used to DDoS the node and should definitely not be exposed). In this case, the P2P ports 9001(tcp/udp) and 30303(tcp/udp). Port forwarding via SSH can be used to access these ports in a secure way.

Redacted: Fixed in PR 56.

Spearbit: The recommendation was followed and a fix was applied in PR 56 at commit 6b5f37a6. Only the P2P ports are now directly exposed directly to the Internet.

5.2 Medium Risk

5.2.1 Docker container running as root

Severity: Medium Risk

Context: dineronode/docker/pirexeth-dockerfile#L1, dineronode/docker/pirexeth-ec-migrator#L1, dineronode/docker/pirexeth-prune-provision#L1

Description: Docker containers run by default as `root`. It is recommended to use a different user to prevent privilege escalation.

Recommendation: Add a `USER` to the Dockerfile. This will require that the files needed to run the node software are owned by that user.

Redacted: Fixed in PR 56.

Spearbit: The recommendation was followed and a fix was applied in PR 56 at commit 6a5de538.

5.2.2 Block depth used does not offer guarantees against reorgs under edge cases

Severity: Medium Risk

Context: cdk-dinero-keeper/src/functions/update-validator-stats/index.ts#L226

Description: The Ethereum chain finalizes roughly every 2 epochs (64 slots), at this point the network offers extreme guarantees for the finalized blocks. The current value of `CONFIRMATION_BLOCKS=30` would be historically safe, but offers no guarantees in edge/attack cases against reorgs and non-finality incidents.

Recommendation: There is a notion of finalized blocks that could be used instead. The data would be older, but would represent the finalized state of the network. Note that in case of a non-finality incident on the network, the value will be stuck in the past unless the network is healthy again. This may even be an advantage as you will not perform actions on data that might change or spend funds that should not have been.

Redacted: Fixed in PR 56.

Spearbit: The recommendation was followed and a fix was applied in PR 56 at commit 3772bdd8.

5.3 Low Risk

5.3.1 Updater misconfigured to check for a non-running container

Severity: Low Risk

Context: dineronode-install/rp-version-check.sh#L5

Description: The current running version is obtained by checking the status of a container named `rocketpool_node`, but such a container would likely not exist. The name being checked should be renamed to ensure the updater works as expected.

Recommendation: Replace `rocketpool_node` with the node name such as `dinero_node`.

Redacted: Fixed in PR 56.

Spearbit: The folder was deleted and the issue addressed in PR 56.

5.3.2 Docker base images should use a SHA256 or a fixed tagged version

Severity: Low Risk

Context: dineronode/docker/pirexeth-dockerfile#L1

Description: The Dockerfile is using `debian:bullseye-slim`. Ideally a fixed version should be used so that the OS version is pinned. It's preferred to use a SHA256 of the image due to the fact that a tag could be re-uploaded. Note that the base image should still be updated from time to time to include the latest security fixes.

Recommendation: Replace `debian:bullseye-slim` with the SHA256 of the `debian:11.8-slim` image.

Redacted: Fixed in PR 56.

Spearbit: The recommendation was followed and a fix was applied in PR 56 at commit 6a5de538.

5.3.3 No security patching process

Severity: Low Risk

Context: Global scope

Description: There's no process in place to automatically update the OS packages regarding security patches.

Recommendation: This is OS dependend, for Amazon Linux it can be something like running`yum update --security` periodically.

Redacted: Acknowledged.

Spearbit: The issue was acknowledged. Instructions will be added to their security playbook to periodically run security updates.

5.3.4 Excessive linux permissions on directories

Severity: Low Risk

Context: linux-ec2-cdk/src/config.sh#L24, linux-ec2-cdk/src/config.sh#L27

Description: There are unnecessary write permissions given to `+g` and `+o`.

Recommendation: Reduce directory permissions from `777` to `755` to avoid other users writing onto those directories.

Redacted: Fixed in PR 56.

Spearbit: The recommendation was followed and a fix was applied in PR 56 at commit c305acf3.

5.3.5 Latest Linux Amazon 2 AMI used without hardening

Severity: Low Risk

Context: linux-ec2-cdk/lib/linux-ec2-cdk-stack.ts#L116

Description: The AMI used could be more hardened.

Recommendation: As an example, a custom AMI could be built that extends the Amazon Linux AMI and runs both `os_hardening` and `ssh_hardening`.

Redacted: Acknowledged.

Spearbit: The issue was acknowledged. Due to the "low" severity, it's planned to be resolved after launch.

5.3.6 AWS Systems Management sessions are not logged

Severity: Low Risk

Context: linux-ec2-cdk/lib/linux-ec2-cdk-stack.ts#L18

Description: The sessiond creates via SSM are not being logged. It's recommended to keep an audit log of these sessions.

Recommendation: Configure a S3 bucket or Cloudwatch log group to log the sessions. The use of encryption for S3 or Cloudwatch using AWS KMS is also recommended.

Redacted: Fixed in PR 56.

Spearbit: The recommendation was followed and a fix was applied in PR 56 at commit 6b5f37a6. The log retention was set to 1 month. It's recommended to keep at least 6 month of logs so that data is available in case of a security breach that is noticed late.

5.3.7 GET `/validators` endpoint on the API Gateway is not cached or throttled

Severity: Low Risk

Context: /cdk-dinero-keeper/lib/cdk-dinero-keeper-stack.ts#L554

Description: The API endpoint that runs the `/validators` function could have some cache with a low TTL to avoid any external abuse of this API endpoint. An abuse could lead to unnecessary read operations and overload on the DynamoDB. The cache is recommended also to avoid increased cloud costs due to excessive lamba invocations and database read operations.

Recommendation: Add a low TTL cache to the endpoint by enabling caching. Additionaly, some throttling limits could be added.

Redacted: Fixed in PR 56.

Spearbit: The recommendation was followed and a fix was applied in PR 56 at commit 7b54deec.

5.4 Informational

5.4.1 Wallet recovery logic is missing

Severity: Informational

Context: dineronode/pirexeth/api/cube3signerwallet/recover.go#L24

Description: The function seems to check if the wallet is initialized and if not then it performs another check to see if it is initialized. This should instead trigger a recovery function rather than a second check.

Redacted: Fixed in PR 56.

Spearbit: The recommendation was followed and a fix was applied in PR 56 at commit 06c34c85.

5.4.2 General notes, performance changes and nitpicks

Severity: Informational

Context: Global scope

Description: Below are a series of general comments to the codebase:

- Lambda function "update-validator-stats" : The beaconcha.in API also provides batch calls. Could be used to reduce API calls.
- Rename `eth1` to `EL` (execution layer) and `eth2` to `CL` (consensus layer).
- Add permissions needed by `GH_TOKEN` during creation to avoid users mistakenly creating a write token.
- DOPPELGANGER_DETECTION is currently not enabled by default. Ideally this feature is enabled on all nodes, for a minor cost in downtime you will obtain extra slashing protection on top of what cubist already offers.
- Convert some of the hardcoded cubesigner values into variables in start-vc.sh#L11. The cubesigner port or the host architecture can change for example.

Redacted: Acknowledged.

Spearbit: Acknowledged.

5.4.3 Cubist policy suggestions

Severity: Informational

Context: Global scope

Description: Cubist allows setting policies that limit the scope of what a session can do.

Recommendation:

- Cubist allows a policy to limit org requests from certain IPs. It might be prudent to setup a company VPN and scoping the sourceIP to only allow cubist requests from the VPN IP + AWS host static IPs. This would greatly help reduce the attack surface in the form of key exfiltration. See index.SourceIpAllowlistPolicy.
- Cubist allows a policy to limit the max unstaked daily at an org level, this could also be prudent to setup and increase only when required. See index.MaxDailyUnstakePolicy.
- After the pre-deposit is made, ensure that the `withdrawal_credentials` is set to a known value before making the full deposit. This recommendation is unnecessary once issue "Cubist policy scoping deposit interactions to known BLS keys attached to role" is addressed.
- Cubist offers a policy to restrict signing transactions to known recipients, this should be scoped to the Deposit contract and pirexeth related contracts and nothing else. This would reduce the blast radius if a cubist token is leaked.
- The cubist organization owner user account should ideally be a cold store account that is never used unless it is an emergency. This is mainly a sane security practice and reduces chances of a token leakage being able to affect the entire organization.

Redacted: Acknowledged.

Spearbit: The issue was acknowledged. Due to the "informational" severity, it's planned to be resolved after launch.

5.4.4 Cubist user session token cleanup

Severity: Informational

Context: dineronode/README.md#L74

Description: Description: Following the guide would leave an (albeit expired) user token created via the `cs login ...` process. This token would have a short lifespan and would likely expire before it can be used, but its presence could allow for an attacker waiting on a compromised host to hijack the session after the user has exited.

Recommendation: Cleanup at the end of the deposit flow/node setup workflow. This would include the `management-session.json` and any other user related key material. The only token left on the machine should be the `signer-session.json` which is used by the cubist `eth2` proxy for BLS remote signing.

Redacted: Acknowledged.

Spearbit: The issue was acknowledged. Due to the "informational" severity, it's planned to be resolved after launch.

5.4.5 Cubist policy scoping deposit interactions to known BLS keys attached to role

Severity: Informational

Context: dineronode/pirexeth/api/node/deposit.go#L136

Description: Cubist policies allow for limiting a role to perform deposits only for the BLS keys attached to the role. This would prevent a scenario in which a rogue machine performs deposits for keys not under the organizations control (see the API docs).

Redacted: Acknowledged.

Spearbit: The issue was acknowledged. Due to the "informational" severity, it's planned to be resolved after launch.

5.4.6 Cubist policy scoping signing key to BLS messages alone

Severity: Informational

Context: dineronode/shared/services/cube3signerwallet/cube3signerwallet.go#L526

Description: Cubist allows for scoping the session token of a role to certain functions. This session being created is used by the `cs eth2 proxy`, whose singular role is to sign BLS messages. The default allows for sign*, which means it can sign any message. Ideally the API calls includes the scope field and it is scoped to purely BLS keys. This would imply that even if the host root access is achieved, the session token would not be able to do any function besides sign BLS messages (reducing the attack surface). API doc can be found here, under scopes.

Redacted: Acknowledged.

Spearbit: The issue was acknowledged. Due to the "informational" severity, it's planned to be resolved after launch.

5.4.7 Cubist token refresh value

Severity: Informational

Context: dineronode/shared/services/cube3signerwallet/cube3signerwallet.go#L513

Description: The cubist signer token is created with a `RefreshLifetime` of `86400`s (1 year). It appears that this creates a file `signer-session.json`, which is used by `cs eth2 proxy`(i.e, the proxy for the remote signer). This would imply that the token would expire in 1 year and the validator would not be able to sign post that timeframe. Its unclear if there is any process in place for refreshing this token, especially one that isn't manual.

Redacted: Acknowledged.

Spearbit: The issue was acknowledged. A new `cs login` has to be performed to create a new session. A process will be in place to know when the session is about to expire so that a manual login can be performed again before the expiration time.

5.4.8 Besu is using sync mode `X_CHECKPOINT`

Severity: Informational

Context: /dineronode-install/install/scripts/start-ec.sh#L254

Description: Checkpoint sync behaves like snap sync, but instead of syncing from the genesis block, it syncs from a specific checkpoint block configured in the Besu genesis file. To avoid trusting this checkpoint configuration, it's recommended to use snap sync.

Recommendation: Switch to `X_SNAP`.

Redacted: Acknowledged.

Spearbit: The issue was acknowledged. Due to the "informational" severity, it's planned to be resolved after launch.

5.4.9 Lighthouse historic state cache increased

Severity: Informational

Context: /dineronode-install/install/scripts/start-bn.sh#L81C11-L81C36

Description: The historic state cache flag `--historic-state-cache-size` was increased from the default value `1` to `2`. This will result in a higher use of memory.

Recommendation: Remove flag and run with defaults (`1`).

Redacted: Acknowledged.

Spearbit: The issue was acknowledged. Due to the "informational" severity, it's planned to be resolved after launch.

5.4.10 Runtime profiling tools enabled in Geth by default

Severity: Informational

Context: /dineronode-install/install/scripts/start-ec.sh#L93

Description: The `--pprof` flag, which enables golang runtime profiling, is enabled. This has a performance impact and should only be enabled when debugging Geth.

Recommendation: Remove the `--pprof` flag or make it configurable when debugging Geth is required.

Redacted: Acknowledged.

Spearbit: The issue was acknowledged. Due to the "informational" severity, it's planned to be resolved after launch.

5.4.11 Dinero node version should use a fixed or tagged version instead of latest

Severity: Informational

Context: dineronode-install/install.sh#L54

Description: The `install.sh` is using `PACKAGE_VERSION="latest"`. Ideally a fixed version should be used so that the release is pinned. Using latest obfuscates the version being used and can lead to config drift. Note that the base version should still be updated from time to time to include the latest security fixes.

Recommendation: Replace `PACKAGE_VERSION="latest"` with the tagged version such as `PACKAGE_VERSION="v0.0.1"`.

Redacted: Acknowledged.

Spearbit: The issue was acknowledged. Due to the "informational" severity, it's planned to be resolved after launch once versioning is in place.

5.4.12 Metrics endpoint exposed globally leaking validating key information

Severity: Informational

Context: dineronode-install/install/scripts/start-bn.sh#L99

Description: The metrics feature of most clients reports the validators registered to said beacon node, especially when used alongside flags such as `--validator-monitor-auto`. In combination with accepting requests from the public internet (`0.0.0.0/0`) would imply that anyone can query the running validator keys on the host, which is information that is ideally not leaked.

- Lighthouse
- Lodestar
- Nimbus

- Prysm
- Teku

Recommendation: Scope the metrics requests to local ranges only, such that a local prometheus can access the data but not external parties. This is prevented by default by the used docker container policies, but would still be prudent to scope to allow for local traffic alone.

Redacted: Fixed.

Spearbit: This issue automatically became fixed due to issue "Open ports to the internet" being fixed. The security group in place doesn't expose the endpoints anymore. The Spearbit team was informed about the multiple layers (security group > docker port forwarding > container process) where ports can be opened and is aware of the risks of opening up ports on the upper layers.

5.4.13 Node Engine-API configured to accept traffic from any IP range

Severity: Informational

Context: dineronode-install/install/scripts/start-ec.sh#L89

Description: The engine API will be open from the internet (`0.0.0.0/0`). The API does offer authentication in the form of JWT token, but there is no limit on the number of attempts. This means the token can be bruteforced in the current setup and would allow anyone to control the `EL` completely. This API can be used to fork the node or to completely stall it.

- Geth
- Nethermind
- Besu

Recommendation: Scope the allowed traffic range to be local only. There should be very few scenarios in which non-local traffic needs to access this API. This is prevented by default by the used docker container policies, but would still be prudent to scope to allow for local traffic alone.

Redacted: Fixed.

Spearbit: This issue automatically became fixed due to issue "Open ports to the internet" being fixed. The security group in place doesn't expose the endpoints anymore. The Spearbit team was informed about the multiple layers (security group > docker port forwarding > container process) where ports can be opened and is aware of the risks of opening up ports on the upper layers.

5.4.14 Node JSON RPC/websocket configured to accept traffic from any IP range

Severity: Informational

Context: dineronode-install/install/scripts/start-ec.sh, dineronode-install/install/scripts/start-bn.sh

Description: The following API will be open from the internet (`0.0.0.0/0`) and would allow anyone to access any API endpoint running. This would lead to a possibility of DDoS attacks and downtime for the node and associated validator. This applies to both the HTTP as well as the Websocket connections.

- Geth-RPC
- Nethermind-RPC
- Besu-RPC
- Lighthouse-RPC
- Lodestar-RPC
- Nimbus-RPC
- Prysm-RPC

- Teku-RPC

Recommendation: Scope the allowed traffic range to be local only. If external HTTP traffic is needed for some reason, please use Nginx or a similar reverse proxy with filtering rules and authentication. This is prevented by default by the used docker container policies, but would still be prudent to scope to allow for local traffic alone.

Redacted: Fixed.

Spearbit: This issue automatically became fixed due to issue "Open ports to the internet" being fixed. The security group in place doesn't expose the endpoints anymore. The Spearbit team was informed about the multiple layers (security group > docker port forwarding > container process) where ports can be opened and is aware of the risks of opening up ports on the upper layers.

5.4.15 Fixed size root EBS volume

Severity: Informational

Description: The VM uses a fixed size (200GB) root volume. To avoid problems with disk space limitations, it's recommended to create a separate volume for data. A separate "data" volume should make it easier to expand the volume in the future without having to touch the root volume.

Recommendation: Have a separate "data" volume and use that volume to store any kind of data that tends to grow (e.g. Geth `data` dir).

Redacted: Fixed in PR 56.

Spearbit: The recommendation was followed and a fix was applied in PR 56 at commit 40792ae. In the same commit, a root volume of 8GB was created. For maintenance purposes, it would be good to increase this slightly, because the OS + logs might quickly fill up the disk space (e.g. 20GB).

5.4.16 IP Address of the EC2 instance could accidentally change

Severity: Informational

Context: linux-ec2-cdk/lib/linux-ec2-cdk-stack.ts#L151

Description: The instance doesn't have an Elastic IP associated. This means that if the instance is ever stopped, it will start up with a new IP address. This could be undesired.

Recommendation: If it's desired to keep the same IP address for the EC2 instance, then an Elastic IP address should be allocated and associated to the instance.

Redacted: Acknowledged.

Spearbit: The issue was acknowledged. The recommendation wasn't applied which means that a changing IP address is acceptable.

5.4.17 `check-unstake` doesn't verify if the broadcast result of the `voluntary_exit` request got included

Severity: Informational

Context: cdk-dinero-keeper/src/functions/check-unstake/index.ts#L260

Description: To exit, a request with a `SignedVoluntaryExit` object is sent to the beacon node via the beacon API. The node should then broadcast this to the network. In extreme cases, it could happen that this message fails to being gossiped across the network and the exit request is never seen.

Recommendation: After doing the request for the voluntary exist, check the validator status periodically `${BEACON_API_URL}/eth/v1/beacon/states/head/validators/${publicKey}` until it's not active anymore.

Redacted: Fixed in PR 56.

Spearbit: The recommendation was followed and a fix was applied in PR 56 at commit 8a07885 and 9804ce86.

5.4.18 Function calling `topUpStake` doesn't wait for transaction receipt

Severity: Informational

Context: cdk-dinero-keeper/src/functions/check-topup/index.ts#L469

Description: The transaction is sent but the function doesn't wait for it to be included. This could give a false sense of a successful function run.

Recommendation: Add `client.waitForTransactionReceipt(...)`.

Redacted: Fixed in PR 56.

Spearbit: The recommendation was followed and a fix was applied in PR 56 at commit f7eca966.

5.4.19 RPC / Beacon endpoints are not checked for correctness

Severity: Informational

Context: cdk-dinero-keeper/src/functions/

Description: Most of the functions use an RPC endpoint from a execution client or a beacon API endpoint from a consensus client. The endpoints are not validated for correctness.

Recommendation: For the RPC endpoint of the execution client, verify if the endpoint is synced (`eth_syncing`) and if the network id matches the desired network (`eth_chainId`). For the beacon API endpoint, validate `/eth/v1/node/syncing` and `eth/v1/config/spec`.

Redacted: Fixed in PR 56.

Spearbit: The recommendation of checking the chain ID was applied in PR 56 at commit dd64697B. It was also acknowledged that the provided RPC endpoints should be in sync, meaning that the monitoring of these RPC endpoints should be done externally to this function.