PDF Export for report 2652784

Failed transaction included in block at ZERO cost

State	Resolved
Reported by	Elliptic Curve (ecc)
Reported to	Cosmos (cosmos)
Submitted at	2024-08-12T17:52:28.480Z (ISO-8601)
Asset	https://github.com/cosmos/gaia (SOURCE_CODE)
References	
Weakness	
Severity	Medium (4.0 ~ 6.9)
CVE IDs	

Summary of Impact

Since v18 of Gaia (specifically introduction of feemarket) it is possible to spam the network with zero cost transactions.

Due to the inclusion of the feemarket fee deduction as a posthandler, if the transaction uses up the entire uatom balance of the wallet, the feemarket handler will attempt to deduct fees, fail to do so, successfully fail the transaction, but fail to deduct fees.

It would be trivial for an attacker to spam the network at ZERO cost, and deny access to legitimate transactions.

Steps to Reproduce

1. Create a MsgSend with the full atom balance of a wallet. The balance sent must be enough to cover the fees that feemarket will require the tx has. However, these will not be deducted.

E.g. wallet balance is 10000uatom, fee is 1000uatom - MsgSend: 10000uatom, with 1000uatom fee.

NOTE: this must not be to self - in this case, once the tx is executed, the account does have the funds to deduct, and the fees are deducted)

Transaction will be included in block, will fail with 'insufficient funds' (feemarket post handler attempting to take fees), and tx will be reverted with zero change in account balance.

To take this further, with 0.125atoms we can set the gas limit to 25m (max allowed by fee market), which will be reverted, having made 1/4 of the block space unavailable. For free.

Workarounds

Not expect not. I believe the issue is related to the inclusion of the feemarket fee deduction as a posthandler.

Supporting Material/References

Zero cost tx:

https://www.mintscan.io/cosmos/tx/0B69B7B4336CC8C1001D988348C02637B67CCAA77E6F1DB5D0B5174880DCB943?height=21706678

zero cost 25m gas tx:
https://www.mintscan.io/cosmos/tx/74C1457AFA93DE2C807891C67F388EBF712A223ED30DE32035CBB5BB18E664E6?height=21707053

Impact

The network can be flooded with transactions that perform zero state changes, for absolutely zero cost. It would cost nothing to make the network unusable for all users by filling block space with zero cost spam.

Activity

2024-08-28 01:34 not eligible for bounty Public

Hi Mo,

The contentious point here is that the fee market module has a bug, but in isolation it is not a risk. The risk is the implementation of that library into a multi billion dollar network. That most definitely is in scope and I’m astonished that

you can claim it is not. I wish to highlight the fact that nowhere in your scope / out-of-scope docs does it state that any third party library bugs are out of scope (which frankly would be an absurd claim, as devs choose what third party libraries to include).

If feemarket were an indirect dependency (dependency of a dependency) I would be able to understand your claim, but this is a direct dependency the devs had complete control over the inclusion of, and should have been picked up in Gaia testing, and thus under your own guidelines is 100% in scope of this program.

2024-08-27 06:18 comment Public

Hey @ecc

As my colleague highlighted, the third party dependency here for Gaia is not covered by the scope in the program, as the scope today focuses on Gaia as a reference implementation of the Cosmos SDK. Additionally, teams outside of this context were contacted about this issue, making this report ineligible for bounty regardless of scope. As program administrators, we're continually working on better defining the scope here, especially with the interop of third party modules and how governance plays a part in adoption of new technologies for the Hub, which can add confusion. In this case with the code owner being Skip, our program isn't scoped to pay out for issues in their codebase.

Best,
Mo

2024-08-27 05:51 comment Public

Thanks Jessy,

Point 11 notes ‘vulnerable libraries’, but explicitly notes ‘… without a specific proof of concept demonstrating the security impact to the in-scope items running in expected configurations.’ which has been provided.

2024-08-27 05:35 comment Public

Hi @ecc,

In this case, the "Out of Scope" designation applies to the component itself and not the class of vulnerability that it represents. As the feemarket source code lives outside of any in-scope property listed in our program scope (it is owned by the Skip Protocol team), it is not eligible for reward.

Thank you again for taking the time to report this issue to the program.

Best,

Jessy

2024-08-27 04:30 comment Public

I’m sorry Mo, but I disagree with this assessment. Point 12 of the out of scope list states:

…DOS attacks that may be mitigated by existing operational controls e.g. gas, fees, etc.

Given that this vulnerability enabled a trivial DOS attack that was not mitigated by fees, it does not appear to be out of scope.

I await your comment.

2024-08-27 04:23 comment Public

Hey @ecc

The Hub team has confirmed this has been addressed in the latest Gaia (v19.1.0). Thank you for reporting this issue through our HackerOne channel. Unfortunately the component is out of scope for the bounty program, so we're not able to issue a bounty, however I will mark the issue as Resolved so this report can contribute to your profile in the HackerOne ranking system.

Best,
Mo

2024-08-27 03:04 bug resolved Public

2024-08-27 03:03	report severity updated	Public

It also impacts Terra and Neturon, as they use the same module.

2024-08-12 19:03	comment	Public

For reference, I have spoken to Barry Plunkett and Tyler at Skip to escalate this directly to them, as they produced the code that is the root of this issue.

2024-08-12 19:03	comment	Public

Hey @ecc

Thanks for your report! We are working with the core team now on a path forward to remediating this case. I will keep you posted as we work through the initial triage process.

Best,
Mo

2024-08-12 18:09	bug triaged	Public